Marcin Skobel

Deep Neural Networks in Medical Image Classification

University of Zielona Góra Press, Poland
2024

Deep Neural Networks in Medical Image Classification

Faculty of Computer, Electrical and Control Engineering
University of Zielona Góra

Lecture Notes in Control and Computer Science Volume 28

Marcin Skobel

Deep Neural Networks in Medical Image Classification

University of Zielona Góra Press, Poland
2024

Marcin SKOBEL
Institute of Control & Computation Engineering
University of Zielona Góra
ul. Szafrana 2
65-516 Zielona Góra, Poland
e-mail: m.skobel@issi.uz.zgora.pl

Supervisor:

- Marek KOWAL, University of Zielona Góra, Poland

Referees:

- Henryk MACIEJEWSKI, Wrocław University of Science and Technology, Poland
- Andrzej POLAŃSKI, Silesian University of Technology, Poland

The text of this book was prepared based on the author's Ph.D. dissertation entitled *Deep neural networks in medical image classification (Głębokie sieci neuronowe w klasyfikacji obrazów medycznych)*.

ISBN 978-83-7842-558-8

DOI https://doi.org/10.59444/2024MONaSko_LNCCS_v28

Camera-ready copy prepared in LaTeX2$_\varepsilon$ by the author.

University of Zielona Góra Press
ul. Podgórna 50, 65-246 Zielona Góra, Poland
tel./fax: +48 68 328 78 64, e-mail: sekretariat@ow.uz.zgora.pl

Printed by the University of Zielona Góra Printing House.

Contents

Acknowledgements **vii**

Notation and abbreviations **ix**

1 Introduction **11**
1.1 Motivation . . . 11
1.2 Work goals . . . 13
1.3 Thesis . . . 15
1.4 State of art . . . 15
1.4.1 Introduction . . . 15
1.4.2 Segmentation . . . 16
1.4.3 Classification . . . 17
1.4.4 Classification on multiple data sets . . . 19
1.4.5 Features fusion . . . 20
1.5 List of the most important achievements . . . 21
1.5.1 General diagram of the classification process . . . 21
1.5.2 Using segmentation to normalize images . . . 21
1.5.3 Deep feature acquisition and feature fusion . . . 21
1.5.4 Selection of diagnostically relevant features . . . 22
1.6 Dissertation structure . . . 23

2 Medical imaging **24**
2.1 Digital image . . . 24
2.2 Datasets . . . 26
2.2.1 A set of cytological images from the Hospital in Zielona Góra 26
2.2.2 BreakHis histopathology image set . . . 26
2.2.3 Data merging . . . 27
2.3 Pre-processing of digital images . . . 30
2.4 Summary . . . 31

3 Image normalization by segmentation **32**
3.1 Introduction . . . 32
3.2 Segmentation using a convolutional neural network . . . 32
3.3 Hybrid segmentation system . . . 34
3.4 Evaluation methods . . . 36
3.5 Verification of the accuracy of the hybrid segmentation method . . . 38
3.5.1 Implementation of a hybrid segmentation method . . . 38
3.5.2 Results of segmentation of cell nuclei . . . 42
3.6 Summary . . . 48

4 Comprehensive classification system **53**
4.1 Introduction . . . 53
4.2 Manual feature extraction . . . 53
4.3 Deep feature extraction . . . 56
4.3.1 Machine learning . . . 56
4.3.2 Elements of artificial neural networks . . . 57
4.3.3 Convolutional neural networks . . . 57
4.3.4 Team learning methods . . . 61
4.4 Dimensionality reduction . . . 63
4.4.1 Extraction and construction of new features . . . 63
4.4.2 Feature selection . . . 64
4.4.3 Stochastic feature selection . . . 67
4.5 Master classifiers . . . 70
4.6 Classification with an artificial convolutional neural network . . . 71
4.7 Evaluation methods . . . 71
4.8 Results . . . 72
4.8.1 Scheme of the empirical research conducted . . . 72
4.8.2 Deep networks . . . 72
4.8.3 Developed classification system . . . 86
4.9 Discussion . . . 104
4.10 Summary . . . 106

5 Summary **107**
5.1 Conclusions . . . 107
5.2 Analysis of results and contribution to the development of the discipline . . . 108
5.3 Further work . . . 109

A Results of additional experiments **110**

B Hardware and software configuration **116**

C U-Net structure for segmentation **117**

Bibliography **119**

Index **128**

List of figures **129**

List of tables **131**

D Streszczenie
Streszczenie **133**

Acknowledgements

I would like to express special thanks to the supervisor of my dissertation, Prof. Marek Kowal, for irreplaceable help in the process of completing my dissertation, transfer of knowledge, commitment and fruitful scientific cooperation.

I would like to express my sincere thanks to Prof. Józef Korbicz for valuable scientific support and indicating the right paths in my work and development, as well as for scientific supervision.

I would like to thank Prof. Dariusz Uciński for enabling teaching cooperation and developing my skills in the area of machine learning.

I would like to thank everyone with whom I had the pleasure of conducting research and publishing results: Prof. Artur Gramacki, Ph.D. Michał Żejmo, M.D.-Ph.D Roman Monczak, Prof. Andrzej Obuchowicz and M.Sc. Norbert Nowicki. I would like to thank Prof. Krzysztof Patan for valuable comments during the presentation of research results.

I would also like to thank my family, especially my wife Basia, my parents Janina and Jan, my mother-in-law Jola and my siblings Justyna, Dawid and Jola. Your support and patience were invaluable, and without your presence and support I would not be where I am.

I would like to express my gratitude to M.A. Dariusz Nowosad for the thorough language correction and consultations.

Notation and abbreviations

V	Variance
sup	Supremum
inf	Infimum
d_{E}	Euclidean Distance
χ^2	Chi-Square
$\bar{X}$	Mean
μ_{I}	First moment
μ_{II}	Second moment
R^2	Coefficient of determination

ACC	Accuracy
AUC	Area Under The ROC Curve
BT	Boosted Tree
CAD	Computer Aided Diagnosis
CCD	Charge-Coupled Device
CMOS	Complementary Metal-Oxide-Semiconductor
CNB	Core Needle Biopsy
CNN	Convolutional Neural Network
DH	Hausdorff Distance
DSF	Dice-Sørensen Factor
DT	Decision Tree
EM	Expectation-Maximization
FNB	Fine Needle Biopsy

GLCM Gray-Level Co-Occurrence Matrix

GLRLM Gray Level Run Length Matrix

GZG Fibroadenoma Zielona Góra

H&E Hematoksylin and eosin

JI Jaccard Index

k-NN k-Nearest Neighbors

LOSS Average data prediction error using a network model

NB Naive Bayes Classifier

NC Neural Classifier

PCA Principal Component Analysis

PNG Portable Network Graphics

RF Random Forest

RL Logistic Regression

ROC Receiver Operating Characteristic Curve

ROI Region Of Intrest

SOB Site Of Biopsy

SSE Sum of Squares Error

SVM Support Vector Machine

SzUZG Dataset from the University Hospital in Zielona Góra

TIFF Tagged Image File Format

TN True Negative

TNR True Negative Ratio

TP True Positive

TPR True Positive Ratio

VSI Virtual Slide Image

Chapter 1

INTRODUCTION

1.1 Motivation

Artificial intelligence and machine learning have become buzzwords which have for years been attracting the attention of researchers and software engineers around the world. New ideas within the field of artificial intelligence have appeared together with the development of relevant equipment. Owing to the symbolic approach, the first intelligent systems appeared in the mid-20th century. The solutions based on the symbolic approach were algorithms made up of a set of rules, Although initially there were high hopes for the symbolic approach (Chollet 2017a), nowadays its methods are being abandoned in favor of sub-symbolic ones, such as artificial neural networks. However, a complete departure from symbolic methods seems to be rather unlikely, as on the one hand, they are still effective and being developed and on the other hand, the use of neural networks is not always fully justified. Also, the symbolic approach includes such advanced methods as genetic algorithms and fuzzy inference.

Nowadays, artificial neural networks represent the landscape of everyday life. They are not a new invention. In the history of science, there had been several instances of their renaissance before they were abandoned en masse in favor of other approaches. The current wave of renewed interest in neural networks emerged around 2012 with AlexNet's (Krizhevsky, Sutskever & Hinton 2012) breakthrough performance in the ImageNet competition. Nowadays, models composed of at least several hidden (deep) layers are known as deep neural networks. Thanks to the intensification of work on algorithms and the rapid technological progress in the building of graphic processors adapted to the needs of machine learning, deep learning possesses further developmental prospects. Moreover, on top of the use of shaders in graphics cards to parallelize neural network calculations, tensor processors are now being designed to ideally answer the needs of deep learning.

The development of deep learning methods also makes it possible to apply them to an increasing number of problems, ranging from the processing of single signals to autonomous vehicles. Among other disciplines, deep learning is also increasingly used in medicine. This work will present some practical applications of deep learning methods to support medical diagnostics, such as segmentation, normalization and classification of medical images.

Correct diagnosis constitutes a key aspect of medical activity and imaging is recognized as one of the most important areas of medical diagnostics. Over

the years of its development, many different medical imaging methods have been developed, including: radiography, ultrasound, magnetic resonance imaging and tomography. The majority of popular medical imaging methods are non-invasive or minimally invasive. In the case of breast cancer imaging, mammography is a commonly used method. It is a preventive, x-ray screening test used for initial detection of cancerous cells.

Its diagnostic scope can be extended to clinical tests, including biopsy. Nowadays, the most commonly used type of biopsy is core needle biopsy (CNB). However, based on the guidelines of the Polish Society of Clinical Oncology (Jassem & Krzakowski 2018), fine needle biopsy (FNB) can also be used in justified cases. The cellular materials used in the experiments included in this work come from FNB and CNB. The advantages of the FNB procedure include its low invasiveness, low costs and refined technology. The disadvantages include slightly lower diagnostic efficiency and the fact that it is currently no longer seen as a gold standard of clinical diagnostic procedures. Despite these disadvantages, the method is still relatively often used in developing countries.

Supporting medical diagnostics is important because there is a shortage of trained specialists, while the number of cancer cases in the world increases from year to year and a continuous increase is predicted for the next 30 years (Pilleron *et al.* 2021). After cardiovascular diseases, cancer is the second most common cause of death in Poland. World statistics estimate that 2,261,419 new cases of breast cancer were detected in 2020, of which 684,996 resulted in death (Sung *et al.* 2021). It is the most common type of cancer among women and accounted for approximately 24.5% of all cancers detected in women in 2020 (Sung *et al.* 2021).

Breast cancer detection involves three stages. The first stage is palpation, after which initial palpation verification and a mammographic or ultrasound examination are performed. Once the presence of a tumor is confirmed, the patient is referred for a biopsy. In this work, experiments were performed on samples from FNB using 0.4 or 0.5 mm diameter needles at the University Hospital in Zielona Gora, as well as on samples from Break-His (Spanhol, Oliveira, Petitjean & Heutte 2016) data from Brazil collected by means of lumpectomy.

Verification of the effectiveness of digital images classification from various research centers, prepared on the basis of various tools, constitutes an additional motivation to deal with this topic. The additional motivation is rooted in critical reports (Maier-Hein *et al.* 2018) regarding biomedical research. The criticism mainly concerns such issues as: inability to compare results, use of various calculation metrics and techniques in different publications, as well as probably the most serious objection regarding achieving better performance through appropriate selection of training and test sets (Maier-Hein *et al.* 2018). Additionally, as results improve, overfitting to the training data may occur, which typically results in reduced model performance on the test data. This leads to difficulties in finding and obtaining digital images of breast cancer biopsies of similar resolution and quality.

The classification of patients into malignant and benign cases based on biopsy images constitutes a considerable challenge for pathologists. The practice of clas-

sifying these images requires at least several years of experience and specialization. In difficult cases, even many years of experience may be insufficient to make an unquestionable diagnosis, which is when it may be necessary to convene a consultation of several specialists and discuss such cases. In addition to scans, doctors also have microscopic samples at their disposal and, above all, the opportunity to perform additional oncological tests.

However, it is encouraging that numerous studies confirm similar diagnostic values of virtual slides and slides under a microscope (Dee *et al.* 2007, Donnelly *et al.* 2013, Evered & Dudding 2011, Gagnon *et al.* 2004, Hanna *et al.* 2017, Van Es 2018). A thorough comparative analysis of diagnostic results based on digital cytopathology and diagnostics based on microscope slides reveals that the obtained diagnoses overlap by 97.6% (Rajaganesan *et al.* 2021). However, research has found (Van Es 2018) that excluding scanning time, diagnoses on virtual slides take more time than microscopic diagnoses. The advantages of digital cytopathology include: storing samples on computers, sharing materials between centers, wide access to materials for teaching purposes for students and doctors undergoing specialization, as well as using digital images to apply convolutional neural networks (CNN) in order to support, diagnose and classify cancers.

1.2 Work goals

CNB has become a standard in the diagnosis of breast cancer, even though FNB is still in use as a less invasive procedure. The diagnostic advantage of CNB has been demonstrated experimentally, i.e. based on data collected from various experiments (Wang *et al.* 2017), it was calculated that the average sensitivity of the diagnosis based on fine needle biopsy (FNB) amounts to 74% (range from 53% to 94%), while the average sensitivity of biopsy (CNB) was estimated at 87% (range from 51% to 96%).

Regardless of the method of material collection, cell nuclei constitute the key element of the diagnosis. Unfortunately, when working on a limited set of training data, preparing training data for segmentation requires a considerable amount of time resulting from the need to manually label cell nuclei in the images. Moreover, with small medical data sets, it becomes necessary to reuse the training data for testing. Therefore, during the experiments performed as part of the dissertation, the k-fold cross-validation method was used in the one-element version (leave-one-out) in order to prevent the leakage of training data into test data. Single-item sets were limited to images within a single patient. The adopted validation method leads to the need to repeat the experiment many times. As a consequence of the adopted assumptions, one of the goals of the work was formulated, which was to build an effective and fast method of segmenting cell nuclei.

The generalization potential of the constructed classifier models is another important problem discussed in this work. In the literature, research almost always involves performing experiments on data from one medical center, which may lead to finding an effective method for classifying images, but from just institution. However, due to the over-fitting of the model to the training data, this method

may prove ineffective on data from another research center. Therefore, another goal set in the dissertation is to **examine the effectiveness of classification based on data from various medical centers**.

Typically, classifiers built on manually extracted features compete with deep learning methods, as the deep network receives input sets of training images in the form of benign and malignant samples, and then, based on the artificially learned features, it acquires the ability to classify images. Unfortunately, this type of approach, referred to in the medical literature as "black-box" (Castelvecchi 2016), does not inspire full confidence in the medical community (Evans *et al.* 2022). However, considering how the classical approach and deep networks perform classification tasks, it can be assumed that combining them into one model may improve the results. Therefore, the next goal of the research was to **verify whether the fusion of manual and deep features would improve the overall classification result**.

Combining manual and deep features requires bringing both types of features to a common dimension. The fusion of manual and deep features also in- creases the total number of features. Sometimes, a rich set of features will require further dimensionality reduction. For this reason, the next goal of the work was to **verify the operation of dimensionality reduction methods and to prepare a feature selection method which is adapted to the nature of the set of features resulting from the fusion**.

As part of their pathology practice, doctors acquire unique skills in the classification of cytological images, but unfortunately they are only able to provide a few key features that are taken into account when classifying a tumor. Therefore, the recognition of other features that influence the classification result is an extremely important task. The input data in the form of images assigned to benign and malignant sets and the lack of clearly defined image features influencing the diagnostic classification determine the use of artificial deep neural networks as the main tool in the feature extraction process.

The paper presents two approaches to the problem of classifying cytological images. The first approach is based on the extraction of deep features resulting from training the network to correctly classify a set of cytological and histopathological images. The second approach starts with semantic segmentation of objects in the image. A diverse set of features is then created based on the segmented cell nuclei. Extraction, which belongs to dimensionality reduction methods, is one of the basic approaches to building a set of features. Thanks to extraction, the original space of the input image is reduced to a set of features describing objects located in the space of this image. Features can be generated in a traditional way, i.e. by determining the morphometric, colorimetric and textural properties of cell nuclei. The set of activities performed in this approach constitutes the comprehensive classification system presented in this work. To achieve the goal, it was necessary to perform detailed research on the verification of the best classifiers, dimensionality reduction methods and the issue of fusion between deep and manual features. The dissertation presents the results obtained in the course of **comprehensive research meant to select the most fitting classification model**.

However, it should be mentioned that there is a set of easy-to-define features that guide doctors when diagnosing malignancy. These include: the size of cell nuclei, disordered arrangement of cells, overlapping of neighboring nuclei, and the ratio of the size of the cell nucleus to the area of the surrounding cytoplasm. Correct diagnoses are determined by these specific features of cell nuclei as well as by doctors experience. **Therefore, another goal of the study was to obtain an accurate classification of breast cancer and find those features that significantly influence its correct diagnosis**.

1.3 Thesis

The features of cell nuclei are obtained from a digital image which constitutes a full feature space. Therefore, extraction of features out of images is one of dimensionality reduction methods. The next step would be to further reduce the dimensionality of the feature space by means of feature selection or projection methods. Classification by means of deep learning additionally involves reducing the dimensionality within the network structure. Based on this information, the following thesis can be formulated:

Improving the classification results of digital cytological images can be achieved by creating a feature space of cell nuclei and reducing the dimensions of this space to a set of features relevant for classification.

In connection to the thesis, three main classification paths will be considered. The first is classic extraction of morphometric, colorimetric and textural features combined with advanced dimensionality reduction methods, in which case, deep networks are used in accurate segmentation of areas of cell nuclei. The second approach is based on the classification of cancer images by means of deep learning methods. Also, a new approach based on the fusion of both classification paths will be proposed.

1.4 State of art

1.4.1 Introduction

The modern surge of interest in classification by means of deep learning began in 2012, when a group of researchers achieved the highest result in an ImageNet competition using the AlexNet neural network (Krizhevsky *et al.* 2012). In 2014, the best result on the ImageNet set was achieved by the GoogLeNet (Szegedy *et al.* 2015) network. Unfortunately, further increase of the network depth led to the phenomenon of gradient vanishing. In 2015, this problem was solved using a ResNet (He, Zhang, Ren & Sun 2016a) network which made use of residual layer technology. Interest in deep learning technology resulted in the publication of deep learning programming libraries such as: Keras (Chollet 2017a), PyTorch (Paszke

et al. 2019), Tensorflow (Abadi *et al.* 2015) and Theano (Theano Development Team 2016).

1.4.2 Segmentation

Doubts regarding the effectiveness of diagnostics by means of digital preparations as opposed to microscopic examination arose with the development of technology for scanning cytological preparations, which makes it possible to view and store samples in digital form. However, later observations revealed similar diagnostic effectiveness of both approaches (Van Es 2018). The ability to effectively diagnose cancer cases based on digital images opens up the potential of using CAD (computer-aided diagnostics) and digital pathology methods.

The application of deep learning is useful in cytology and histopathology image processing in two key areas. The first is segmentation of cytological images, with particular emphasis on the detection of cell nuclei. However, there are alternative segmentation methods involving morphological transformations or stochastic algorithms. The simplest method of segmenting cell nuclei is thresholding, which offers many approaches (Hayakawa *et al.* 2021), including: adaptive thresholding methods (Lu, Mahmood, Jha & Mandal 2012), adaptive thresholding with modeling the shape of the cell nucleus (Phoulady *et al.* 2016), thresholding and smoothing (Fukuma *et al.* 2016), centroid algorithm with thresholding (Anishiya & Sasikala 2016), thresholding with EM algorithm (Phoulady, Goldgof, Hall & Mouton 2016) or optimal thresholding with hierarchical mean shift (Hou *et al.* 2016). With the use of appropriate pre-processing, segmentation based on thresholding can be highly efficient.

Insufficient separation of individual objects in the process of instance segmentation is a common problem when making use of thresholding methods. Marker-controlled watershed segmentation (Veta *et al.* 2011, Veta *et al.* 2013, Cui & Hu 2016) constitutes a solution to both under- and over-segmentation. The process of generating markers may itself be preceded by morphological operations (Veta *et al.* 2011, Salvi & Molinari 2018) as well as by other methods, e.g. initial binarization of the cell nuclei area and determination of erosion points (e.g. Ultimate Erode Points) (Shu *et al.* 2013). Other approaches include watershed segmentation with Otsu thresholding and Hough transformation (Rajyalakshmi, Rao & Prasad 2017) and watershed combined with hierarchical centroid algorithm (Shi, Zhong, Huang & Lin 2016). Marker-controlled watershed segmentation continues to be popular and, under the right conditions, can be extremely effective, i.e. exceeding 91% F-score accuracy (Salvi & Molinari 2018). Segmentation methods based on morphological operations in combination with other techniques such as: thresholding (Petushi *et al.* 2006, Win & Choomchuay 2017), centorid algorithm (Neghina *et al.* 2016, Zarei *et al.* 2017), PCA (Tareef *et al.* 2016) or the Hough transformation (Ragothaman, Narasimhan, Basavaraj & Dewar 2016) are also very popular.

Another group of approaches that have been popular in recent years are active contour methods. The idea of the method is to build a boundary of the object using spline functions and then to transform them in such a way as to minim-

ize the value of the energy function defined by gradient information (Hayakawa *et al.* 2021). Examples of the use of active contours include the geodetic active contour controlled by the EM algorithm (Fatakdawala *et al.* 2010) (sensitivity 0.80, precision 0.86) and active contours with a set of levels based on an interactive model (Qi, Xing, Foran & Yang 2011) (sensitivity 0.78, precision 0.90).

Approaches based on cluster analysis using the centroid method have also been used in medical image segmentation tasks. Centroid clustering returned segmentation efficiency of histopathological images from a TCGA (*The Cancer Genome Atlas*) set at the level of 97.6% accuracy (Niazi *et al.* 2017). Among the proposed solutions, fuzzy centroid clustering (Saha, Bajger & Lee 2016) returned effectiveness in segmenting cytological images from a ISBI 2014 set, at the level of 0.933 precision and 0.929 sensitivity.

Currently, methods based on deep learning have become the "gold standard" in digital medical image segmentation tasks. The use of neural networks can lead to effective semantic segmentation of objects in an image. Unfortunately, semantic segmentation alone may not be sufficient if the key task is to obtain single objects, i.e. instance segmentation. One possible solution is to introduce a third class of objects in the form of object boundaries, which made it possible to obtain breast cancer cell nuclei segmentation results from 0.7478 to 0.9149. In the case of segmentation by means of artificial neural networks, U-Net networks performed particularly well (Alom, Yakopcic, Taha & Asari 2018, Cui *et al.* 2019, Naylor, Laé, Reyal & Walter 2019, Ronneberger, Fischer & Brox 2015). Connections with the use of preliminary segmentation using a CNN network supported by a watershed algorithm and with the detection of watershed markers using a U-Net neural network were also considered.

In the literature, recent research work focuses on extending the U-Net structure with known deep network architectures (Lagree *et al.* 2021). U-Net is a two-part encoder-decoder network, where the role of the encoder can be played by a very deep network model (e.g. VGG, ResNet, Xception). These properties were examined for the quality of medical image segmentation (Lagree *et al.* 2021, Zhang, Du, Xiao & Liu 2020, Zhang, Wu, Coleman & Kerr 2020). In parallel, segmentation technology using region-based CNNs (R-CNN) is being developed. Networks in the *Mask R-CNN* (He, Gkioxari, Dollár & Girshick 2017) version may be particularly useful for cytological images, because their task is to segment object instances. The article (Lagree *et al.* 2021) also proposes a set of U-Net segmentation classifiers built on the basis of VGG-19 (Simonyan & Zisserman 2015), DenseNet-121 (Huang, Liu, Maaten & Weinberger 2017) and ResNet-101 networks (He *et al.* 2016a). Research conducted around the world (Lagree *et al.* 2021) reveals comparable effectiveness of U-Net classifiers, Mask RCNN and the use of a team of classifiers at the input to U-Net. The obtained average accuracy of the Jaccard Index detected after the segmentation of cell nuclei ranged from 0.53 to 0.54.

1.4.3 Classification

The development of medical imaging contributed to the acquisition of high-quality digital cytological images already in the early 1990s and the performance of ef-

fective classification (Street, Wolberg & Mangasarian 1993). As a consequence of the experiment carried out at that time (Street *et al.* 1993), an approach was used consisting of segmentation of cell nuclei, obtaining morphometric, colorimetric and textural data and then performing classification on the extracted data. Further work (Mangasarian, Street & Wolberg 1995, Wolberg, Street & Mangasarian 1994) led to obtaining a data set containing several dozen extracted features. This kit, known in the literature as Breast Cancer Wisconsin, continues to be the basis of current scientific research (Kumar *et al.* 2020, Naji *et al.* 2021, Sakib *et al.* 2022, Solanki *et al.* 2021, Yedjou *et al.* 2021).

The following years resulted in the development of a huge number of different solutions in the field of cytological and histopathological image processing using neural networks, including issues related to classification. For this reason, the topic of systematization of existing solutions was undertaken. One of the systematizations of deep learning models, proposed in the literature, classifies various methods into 4 main groups: supervised learning, semi-supervised learning, unsupervised learning and transfer learning. Examples of supervised learning are typical classification models using CNN (He *et al.* 2016a, Veta *et al.* 2013). Among the semi-supervised methods, their application can be found in the literature on the classification of histopathological images of breast cancer and colon cancer (Ilse, Tomczak & Welling 2018) and in the classification of histopathological images of lung cancer (Wang *et al.* 2020). In turn, an example of the use of unsupervised learning in the process of classifying histopathological images, specifically virtual slides, is an article (Bulten & Litjens 2018) discussing cases of prostate cancer tumors. In the literature (Chennamsetty, Safwan & Alex 2018) you can also find an example of breast cancer classification in histopathological images using transfer learning and using the voting method of an ensemble of CNN-based classifiers. The article reports a result of 87% accuracy. Another example of the use of transfer learning on histopathological images of breast cancer is the article (Kwok 2018) using the deep InceptionResNetV2 network for this purpose, reporting the accuracy of 87%.

Supervised learning methods can be divided into two main approaches. The first ones are based on the extraction of image features. The classifier then takes these features and makes a diagnostic decision (Fondón *et al.* 2018, Jeleń, Fevens & Krzyżak 2008, Kowal *et al.* 2013, Kowal, Skobel & Nowicki 2018, Kowal, Skobel, Gramacki & Korbicz 2021, Naji *et al.* 2021, Sakib *et al.* 2022). The second approach uses a CNN network to classify images (He *et al.* 2016a, Spanhol *et al.* 2017, Veta *et al.* 2013). The dissertation discusses the issue of supervised learning in both approaches and examines the effect of fusion of these approaches.

In approaches based on supervised learning, a support vector machine (SVM) is often chosen as an effective classifier. Slightly worse results are obtained for other classifiers: decision trees (DT), k-nearest neighbors (k-NN), naive Bayes classifier (NB) (Fondón *et al.* 2018, Kowal *et al.* 2013, Kowal *et al.* 2018). Publications (Fondón *et al.* 2018, Kowal *et al.* 2018) showed an average classification accuracy of 75-76%, but under favorable conditions the classifiers exceeded 80% accuracy. It can therefore be assumed that it is almost impossible to exceed 90% accuracy

using standard classifiers. The results of classification based on deep features look slightly better (accuracy 81-86%) (Spanhol *et al.* 2017), as well as classification using the CNN network (85-90%) (Veta *et al.* 2013) .

Nevertheless, it would be extremely interesting to build a classifier that would be similarly effective not only in data from one research center, but also from other institutions. In other words, the work will involve finding features in medical images that will become the basis for building a system that will effectively classify both data from the same medical facility, but also external data. Therefore, the experiments will use images from the University Hospital in Zielona Góra and the Laboratory of Anatomy, Pathology and Cytopathology in the city of Parana, Brazil. These collections differ in almost everything (sampling method, technology, equipment, geographical location). The similarities between the data are that they visually represent images of breast cancer cells and have been treated with hematoxylin and eosin (H&E).

1.4.4 Classification on multiple data sets

Most often, in the literature, you can find experimental studies based on one set of data (from the same research center). Unfortunately, for a medical diagnostic support system to be fully effective, it should demonstrate classification accuracy for various data. One of the studies on the effectiveness of the classification of the same method on different datasets (Herlev (Jantzen, Norup, Dounias & Bjerregaard 2005), HEMLBC (Zhang *et al.* 2014)) involved the application of a deep neural network with transfer learning, where the initial weights were derived from the ImageNet set, while the network was further trained on the studied datasets (Zhang *et al.* 2017). The study was performed on cytological images of cervical cancer obtained from various smear collection methods. The classification involved the detection of normal and abnormal cell nuclei. The accuracy result obtained on both sets was 98.3%. Another study (Sornapudi *et al.* 2019) used two different datasets (their own data and the Herlev (Jantzen *et al.* 2005) dataset). After initial data processing, one common set was created, from which training, validation and testing data were separated. In the next step, the classification efficiency of 4 different deep networks was verified, the best result of which was obtained on VGG19 (Simonyan & Zisserman 2015). The accuracy of the obtained model was 0.8871. In the literature, you can also find works (Arooj *et al.* 2022) regarding the classification of breast cancer on up to 3 data sets. Each of the acquired sets was used to create three separate training and test data sets. These sets were used to build classifiers based on the AlexNet (Krizhevsky *et al.* 2012) network using transfer learning. Depending on the set, the proposed method achieved accuracy from 96.7% to 100%. Therefore, it can be noticed that in the literature, authors focus on testing the proposed method on various data sets within these sets or combine the sets into one set and train and test the model on these data. Therefore, the question remains what happens when data from one medical center is used to train the model and data from another center is used to check the accuracy of the obtained model.

1.4.5 Features fusion

One of the most important reasons for using deep CNNs is their ability to independently select diagnostically important image elements. This feature eliminates the time-consuming process of feature engineering. CNNs are therefore an alternative to manual feature extraction methods. This does not mean, however, that the image features generated by the deep network have their equivalents among manual features. Moreover, it can be said that deep features are usually abstract and there is little probability of their convergence to manual features. Therefore, there is a suspicion that the fusion of manual and deep features will improve classification results.

The idea of fusion of deep and manual features has been successfully used in the problem of classifying plants by leaves (Hall *et al.* 2015). In this study, a deep feature generator in the form of a ConvNet network and a group of manual features based on the shape and color of leaves were used. Classification results from both groups of features were generated using random forest (RF). The final result showed an improvement in the classification result on the created fusion of classifiers by 6.1% compared to the classifier based on manual features and by 2.8% compared to the result obtained using the classifier based on deep features.

Research on the effectiveness of classifiers based on sets of manual and deep features in histopathological issues indicates that the highest classification results are obtained for classifiers based on deep features and comparable results for classifiers based on the fusion of deep and manual features (Tripathi & Singh 2020). This study used the extraction of deep features from dense layers of various neural networks. These layers had the number of neurons from 1024 to 4096, giving a total of 17,408 deep features. This number was reduced using the PCA method. The classification of four types of cancer was performed on the CRCHistoPhenotypes (Sirinukunwattana *et al.* 2016) set, which has 100 cancer cases. The authors of the study (Tripathi & Singh 2020) report obtaining a classification result of 0.9981 for the set of deep features and 0.9978 for the fusion of deep and manual features. Based on existing research, it can be concluded that the fusion of deep and manual features can improve classification results or, in the worst case, maintain the level of the better of the two groups of features.

Fusion of manual features with CNN networks have been used in the task of detecting tumor cell nuclei in histopathological images (Kashif *et al.* 2016). This study showed an improvement in the accuracy (F-score) of cell nuclei detection from 0.709 for the CNN network to 0.748 for the method combining manual features with the CNN network. Another example of combining manual features with CNN networks is the detection of objects (cell nuclei during mitosis) in histopathological images (Wang *et al.* 2014). In this study, the accuracy (F-score) of mitotic change detection was 0.5730 for CNN, 0.6864 for manual features, and 0.7345 for combined approaches.

1.5 List of the most important achievements

1.5.1 General diagram of the classification process

The approach used is a complex set of activities leading to obtaining the most effective model for classifying medical images of breast cancer. The whole process (Fig. 1.1) can be divided into three main parts: image segmentation, creating a feature space and selecting the most important ones, and image classification.

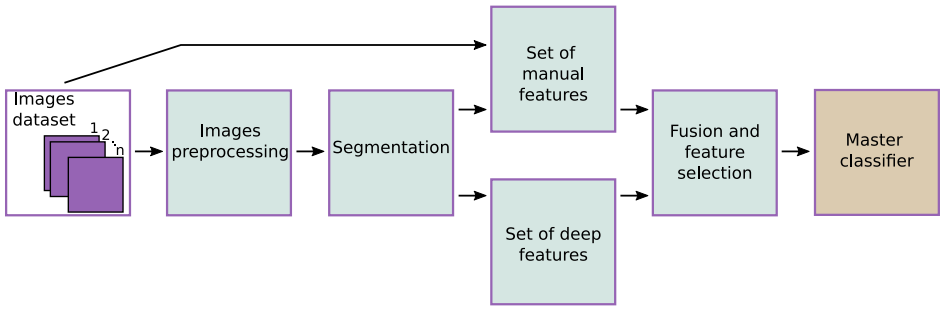


Figure 1.1: Diagram of the proposed approach

1.5.2 Using segmentation to normalize images

The first is image segmentation, which, in addition to semantic segmentation itself, still has the task of normalizing images before feeding them to the input of CNN-type classification network models. The need to develop such normalization stems from the lack of generalization capabilities of networks trained on images in RGB space. The last important effect of using segmentation is the rejection of irrelevant images in which cell nuclei are not present at all or are very sparse. The segmentation method developed within the scope of this work consists of a hybrid system based on CNNs and a watershed algorithm. Segmentation is devoted to a chapter 3 of this dissertation.

1.5.3 Deep feature acquisition and feature fusion

The second part of the process involves manual feature extraction and deep feature generation. Feature extraction firstly requires performing effective segmentation of instances of cell nuclei. Single-cell nuclei extracted in this way in binary form constitute the basis for calculating the morphometric features of objects, as well as masks cutting out interesting parts of more complex images. The generation of deep features, on the other hand, is due to the fact that during experiments it was proven that models learned from segmented images have higher generalization abilities than models learned from images in RGB space. A modified concept of classifier ensemble was used to generate a set of deep features (Fig. 1.2). The modifications made to the original algorithm consist of building homogeneous input samples for the classifiers. These samples consist of a set of learning and

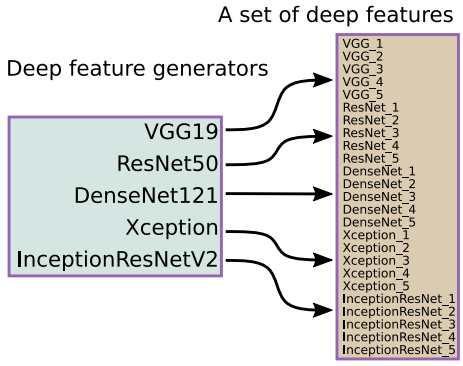


Figure 1.2: Deep feature acquisition scheme

validation data divided in the same way, while the models of the classifiers differ. This procedure makes it possible to obtain different deep feature values from the prepared data sets from different neural network architectures. The final step of the algorithm is to remove the last dense layer in the built neural network models and extract the values of the neuron outputs from the penultimate layer. These values are then used to build deep feature vectors and then manual and deep feature fusion. For a detailed description of the issue, see subsection 4.3

1.5.4 Selection of diagnostically relevant features

After obtaining a set of deep, manual features and also performing feature fusion, the total number of features was 276. As a result of the study, the highest feature fusion classification result was obtained for regression methods.In order to examine what number of features give the best classification results, an experiment was conducted in which a number and a random set of features limited by this number were drawn. After repeating the experiment multiple times, the top 100 results were selected along with information on what number of features were needed to produce the best classification result. This draw represents the first stage of the proposed feature selection algorithm. The experiment proves that the empirical distribution is close in shape to the Gamma distribution (Fig. 1.3).

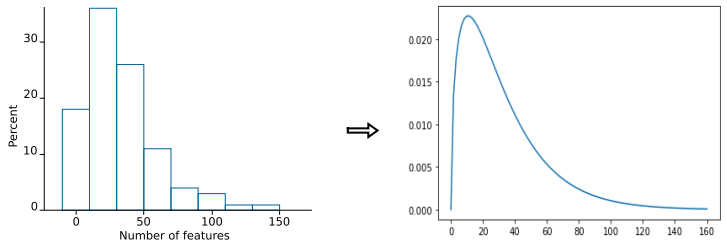


Figure 1.3: Empirical distribution and fitted Gamma distribution

After estimating the values of the parameters of the Gamma distribution, the stochastic algorithm begins. Inside the loop, the length of the feature vector is selected and a set of features limited by a previously randomly selected number is drawn. The obtained feature vector is the input data set for building the logistic regression model. To avoid over-fitting the model to the learning data, cross-validation of the learning data and L2 regularization were applied. For a detailed description of the method, see subsection 4.4.3.

1.6 Dissertation structure

The work consists of five chapters. The first chapter is an introduction to the issues discussed, i.e. an introduction to the topics discussed in the work and the motivation to take up the problem, then the purpose of the work is discussed. The introduction also includes the thesis of the work, the current state of knowledge, the main achievements of the work and a discussion of the content of the dissertation. The second chapter deals with medical imaging and the equipment used to acquire cytological images. The chapter also includes a description of the dataset and the source of the cytological images, and addresses issues related to their preprocessing. The third chapter deals with the issue of image segmentation, with particular emphasis on deep learning methods. The chapter describes the segmentation method developed and the results obtained. The fourth chapter, in turn, deals with the issue of classification of cytological and histopathological images both classically and with the application of deep learning techniques. In addition, the chapter presents a comprehensive method for effective classification of breast cancer images from various medical centers. The theoretical introduction is followed by a section with empirical results along with a discussion and summary. The last chapter is a general summary of the dissertation's content and an analysis of the results for future work.

Chapter 2

MEDICAL IMAGING

2.1 Digital image

A raster digital image is composed of pixels with numerical values from a specific range. Pixels are defined in image space using coordinates. A digital image can be acquired by recording it using a CCD or CMOS photoelectric converter. CMOS sensors are popular in multimedia devices and hand-held digital cameras, while CCD matrices are more often used in photogrammetry and medical imaging. Unlike medical samples, which lose their biochemical properties over time, a properly stored as a digital image is a permanent carrier of the data it contains. Obtaining digital images of biopsy samples is possible thanks to the use of a virtual slide scanner with a CCD matrix. A virtual slide is a digital scan of a recorded sample. The material consists of a fragment of human tissue. In cytological imaging, it is essential to use high-quality images, because the color, texture or shape of objects in a digital image may be important in diagnostics. The TIFF format, along with uncompressed images in the PNG format, are standard formats for working with digital images in the presented experiments.

The images that are the subject of the research included in this work were taken by means of an Olympus VS120 device. The device scans images into a closed VSI format, which is supported by the OlyVia browser. An alternative method to open and edit a VSI file is to use a Bio-Formats library (Linkert *et al.* 2010). A VSI file consists of many layers with different levels of detail (resolution) of stored images. Compared to an image with maximum resolution, this format is an effective solution to the problem of limited RAM resources. Namely, taking into account the fact that the resolution of an image at full magnification is 100,000 by 200,000 pixels, we can calculate that the image contains 2×10^{10} pixels, and after taking the RGB color space into consideration, we end up with a total of 2×10^{10} pixels. When converted into memory units, this value amounts to 55.88 GB, which is above the standard RAM size of modern PC computers. It should also be noted that there are only a small number of formats available for storing such files without compression (e.g. BigTIFF). The scanner used in the research features a forty-times optical magnification lens and a very sensitive CCD matrix with a pixel size of 3.45 μm. Thanks to these properties, it is possible to very precisely see the structure of the cells contained in the preparation (Fig. 2.1). Maximum magnification makes it possible to isolate characteristic fragments of cell nuclei, such as the nucleolus, cell membrane, cytoplasm and other organelles. Addition-

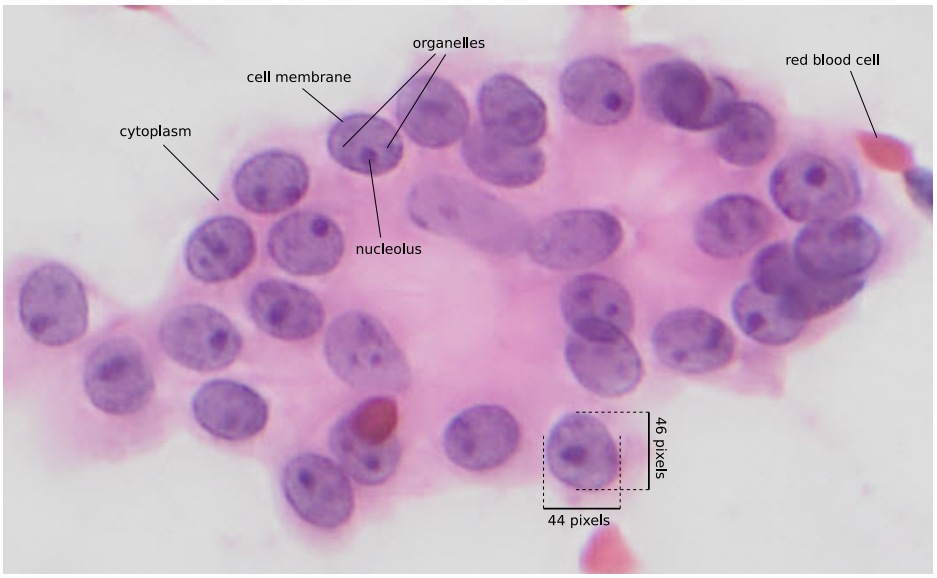


Figure 2.1: Description of the specimen at maximum magnification (benign case)

ally, red blood cells can be seen on the preparations, which are a valuable source of diagnostic knowledge for doctors. Red blood cells are always similar in size, which in turn makes it possible for the assessing physician to determine the size of the cell nucleus in relation to the size of the red blood cell. Computer verification of the size of a cell nucleus does not require the presence of a red blood cell in the image, but only the correct segmentation of individual cells. It is therefore the size of cell nuclei that becomes one of the first features to which diagnosticians pay attention.

There are also other factors that determine the final result in the form of a digital cytological image. Biopsy procedures begin with the correct location of the puncture site. The act of inserting the needle into the tumor is supported by ultrasound imaging, but it should be noted that it is best to use supporting imaging from at least two perpendicular sides. This significantly increases the probability of hitting the tumor. The subsequent steps involve staining the sample, correct positioning and making an appropriate smear on the slide. What is most interesting is the fact that the preparation is stained by means of acidophilic eosin (stains basic fragments) as well as acidic hematoxylin, which stains (basic-absorbing) cellular structures. Eosin gives red color to sample fragments, while hematoxylin is a blue dye. The effect of these two coloring substances is illustrated in Fig. 2.1. The color of the sample is important for further digital image processing.

2.2 Datasets

2.2.1 A set of cytological images from the Hospital in Zielona Góra

The collection of cytological breast cancer images was created as part of cooperation between the Institute of Control and Computation Engineering at the Faculty of Computer, Electrical and Control Engineering of the University of Zielona Gora and the Department of Pathomorphology at the University Hospital in Zielona Gora. The research material collected as part of the project consists of a database organized according to the severity of the cases.

The data used for the experiments were obtained by pathologists from the Department of Pathology of the University Clinical Hospital in Zielona Góra. As a result of fine-needle biopsy, 25 virtual preparations containing benign cases and 25 preparations containing malignant cases were obtained. Diagnostically significant fragments of the virtual slides were marked by doctors on 11 images for each patient, giving a total of 550 images. Some of the images were subjected to manual segmentation (into cell nuclei, cytoplasm with red blood cells and background). In addition to the two sets mentioned above, a set of 25 virtual slides of fibroadenomas cases was also obtained. As in the previous sets, doctors selected 11 diagnostically significant images for each slide. The acquired data were used to build and train a CNN network for automatic segmentation of cytological images.

It is worth emphasizing that the number of patients is unpredictable and difficult to obtain in significant numbers. For this reason, 50 patient samples were obtained for the experiment. According to the literature (Chollet 2017a), the use of an artificial neural network requires a large amount of training data. The number of 50 patients, from which a training and test set should be separated, therefore seems to be small. However, this deficit is partially compensated by the size of the virtual slides. The area of a single virtual slide is approximately 200,000 by 100,000 pixels. Additionally, the virtual slide has several less detailed layers (series) used to smoothly view the content of the virtual slide. An 800 by 1,500 pixel image occupies an almost imperceptible part of the virtual slide (Fig. 2.2).

2.2.2 BreakHis histopathology image set

Referring to information obtained from the website of the Federal University of the State of Parana in Curtiba (Spanhol *et al.* 2017), the BreakHis collection of 9109 digital images taken from a group of 82 patients. Unfortunately, most of the images in the database are low magnification photos. Only the set marked as "400X" contains images corresponding in scale to the images in the SzUZG set. This is due to the fact that a microscope with a forty-fold optical magnification lens was used to obtain images in both cases. It is important to distinguish optical magnification from visual magnification, because the BreakHis collection is recognized by the name "400X" due to the visual magnification of objects by 400 times. The above explanation leads to the final statement that the BreakHis "400X" set contains 1820 images divided into two main groups, i.e.: benign in the number of 588 images and malignant, the number of which is 1232. These data were created

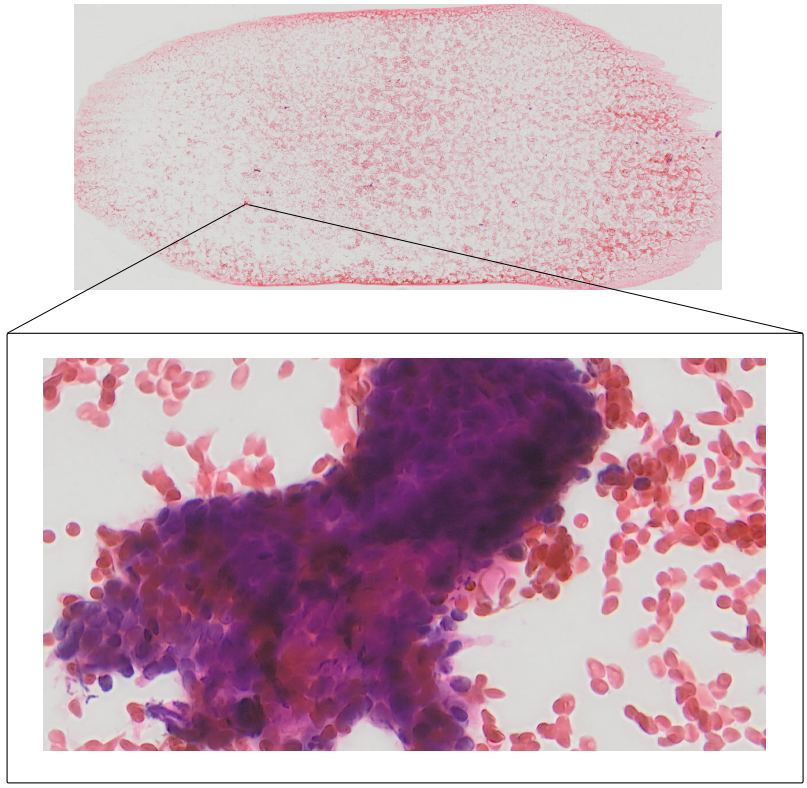


Figure 2.2: Maximum magnification of a portion of the virtual slide area

based on 82 patients. The BreakHis collection is additionally characterized by a detailed division of the benign collection into the following groups: adenoma (*Latin: adenosis*), fibroadenoma (*Latin: fibroadenoma*), phyllodes tumor (*Latin: phyllodes tumor*) and tubular adenoma (*Latin: tubular adenona*). In turn, malignant cases were divided into the following groups: cancer (*Latin: carcinoma*), lobular carcinoma (*Latin: lobular carcinoma*), mucinous carcinoma (*Latin: mucinous carcinoma*) and papillary carcinoma (*Latin: papillary carcinoma*). The images have a resolution of 700 by 460 pixels, have three channels (RGB) with 8-bit depth in each channel and are saved in the PNG format. Samples were obtained using site of biopsy (SOB).

2.2.3 Data merging

Shortage of patients is the most important problem related to the processing of digital medical images of cancer tissue. For example, the SzUZG set has 550 images but only 50 patients. Therefore, it was necessary to acquire an additional database to supplement the existing resources and verify the effectiveness of the proposed methods on data from various centers.

Databases from different centers present the first problem, i.e. the discrepancy in image resolution and different formats (PNG and TIFF). Because like PNG, TIFF is a lossless format, both are suitable for medical images. Ultimately, due to smaller file sizes and slightly better support of Python libraries, the PNG format was chosen for the experiments.

While analyzing the content of the BreakHis X400 collection, a significant disproportion in the number of benign (588) and malignant (1232) images was noticed. This problem was solved by adding a set of fibroadenomas from the database to the BreakHis data. The new set was labelled as BreakHis + GZG, where GZG stands for fibroadenomas from the University Hospital in Zielona Gora. The difference in image resolutions (SzUZG - 1583 $\times$ 828, BreakHis - 700 $\times$ 460) constituted another file compatibility problem. The problem was solved in such a way that from each image of size 1583 $\times$ 828, 5 images of size 700 $\times$ 460 were cropped, one of which was cut from the very center and four in the corners (Fig. 2.3). In

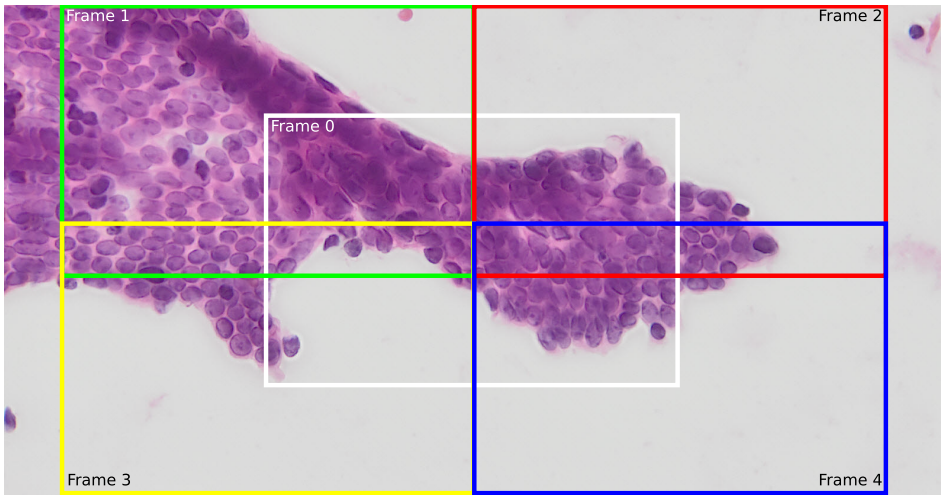


Figure 2.3: Crop the image to 700 $\times$ 460

addition, the cropped images were subjected to preliminary selection, which involved rejecting images dominated by the background, the rejection criterion being an above 20% coverage of pixels belonging to cell nuclei. Pixel coverage was made based on fast binarization using image deconvolution (Ruifrok & Johnston 2001) and then binarization using the Otsu (Otsu 1979) method. This solution is not as accurate as binarization using artificial neural networks, but it is extremely fast and does not require the use of a training set. These are the two most important criteria that must be met by initial binarization performed on a significant number of images. Ultimately, adding the fibroadenomas set to BreakHis resulted in a ratio of benign to malignant images at 53% (1408) to 47% (1232).

In order to match the BreakHis set, SzUZG, i.e. the second image database, was cropped in the same way as the GZG set. Similarly, deconvolution and binarization were applied to detect areas occupied by pixels belonging to cell nuclei, and images containing a significant amount of background were rejected. In this case, however, the rejection criterion was slightly less stringent and amounted to 10% coverage. Thanks to this, a comparable number of images was obtained in both databases and an excellent ratio of the number of benign (1318) to malignant (1330) images at approximately 50/50 was established.

Finally, the prepared images were used to perform manual segmentation. Then, the manually segmented images were used to build an automatic segmentation algorithm by means of CNN. The details concerning segmentation methods will be discussed in the following sections. Based on 700 × 460 images, 230 × 230 images were cropped, which were then used to train classifying artificial neural networks. Due to the high complexity of the image preparation process, the entire procedure was additionally supplemented with schematic drawings (Fig. 2.4). Therefore, the basic classification unit is an image with a size of 230 × 230 pixels, which is important because research (Kowal *et al.* 2018, Skobel, Kowal & Korbicz 2020) has shown that the classification of single cell nuclei achieves real accuracy of 80%. Hence, the conclusion that further research should be based on larger data units, including images containing a larger number of cell nuclei.

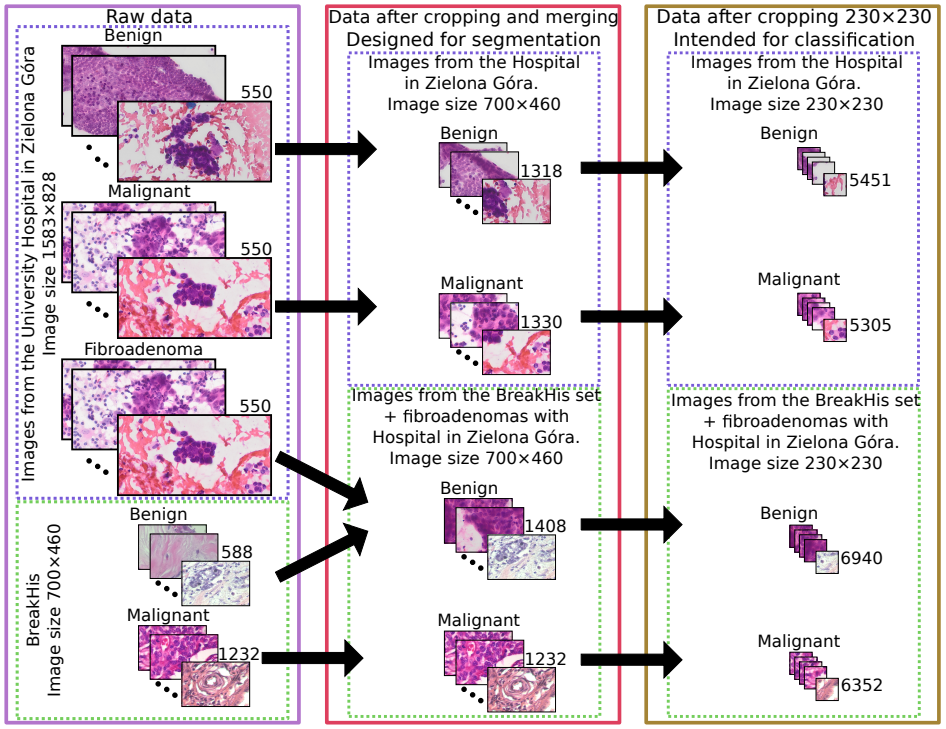


Figure 2.4: Scheme for preparing databases with images

Cropping the images to 230 × 230 was dictated by the size of the images in the BreakHis set, which is 700 × 460. When cropping, you can generate 6 unique 230 size frames from one image with two 5-pixel wide strips cut off on the right and left sides. Once again, the criterion for rejecting images containing too much background was used. In the case of the SzUZG set, the criterion was 7.5% pixel coverage of cell nuclei, while in the case of the BreakHis set, the criterion was 19%. The thresholds were selected experimentally in such a way as to optimally distribute the ratio of the number of malignant to benign cases in individual sets and at the same time balance the total number of images in both sets. Ultimately, 10756 (45% of all) images were obtained in the SzUZG set, including 5451 (51% of all) benign and 5305 (49% of malignant) images, while in the BreakHis + GZG set the number amounted to 13292 (55% of all), including 6940 (52%) benign and 6352 (48%) malignant.

2.3 Pre-processing of digital images

The above-mentioned image pre-processing plays an extremely important role in working with digital images. One of the most popular pre-processing methods is image deconvolution. In the case of digital images of biopsy samples, deconvolution helps to initially separate cell nuclei from the rest of the objects in images. This is possible thanks to the use of a concept based on the law of absorption (Lambert-Beer) (Ruifrok & Johnston 2001). The HistomicsTK library (*https://pypi.org/project/histomicstk/*) was used to perform the deconvolution operation. The samples were fixed using acidophilic eosin and basophilic hematoxylin, therefore a deconvolution matrix was used in the following form:

$$\begin{bmatrix} eosin: \\ hematoxylin: \\ null: \end{bmatrix} \begin{bmatrix} 0.07, 0.99, 0.11 \\ 0.65, 0.70, 0.29 \\ 0.00, 0.00, 0.00 \end{bmatrix} \tag{2.1}$$

After deconvolution, the resulting image can be normalized or further processing can be started. Depending on the chosen processing path, the subsequent stages may include the use of morphological operations or binarization. Ultimately, the image is subjected to semantic segmentation. The effect of applying deconvolution is presented in Fig. 2.5. Deconvolution is an introduction to further processing of cytological images. In the case of image pre-processing used in the described experiment, the Otsu (Otsu 1979) binarization method was used. This method is classified as global, which means that the entire image is used to determine the binarization threshold. The function of the method is to analyze the histogram as well as to minimize the intra-class weighted sum of both classes:

$$V_w(b) = w_0(b)V_0(b) + w_1(b)V_1(b), \tag{2.2}$$

where w_0 and w_1 are the probabilities of both classes, and b denotes a specific value of the pixel intensity threshold. V_0 and V_1 are the variance values. Minimizing the intra-class weighted sum is equivalent to maximizing the inter-class variance.

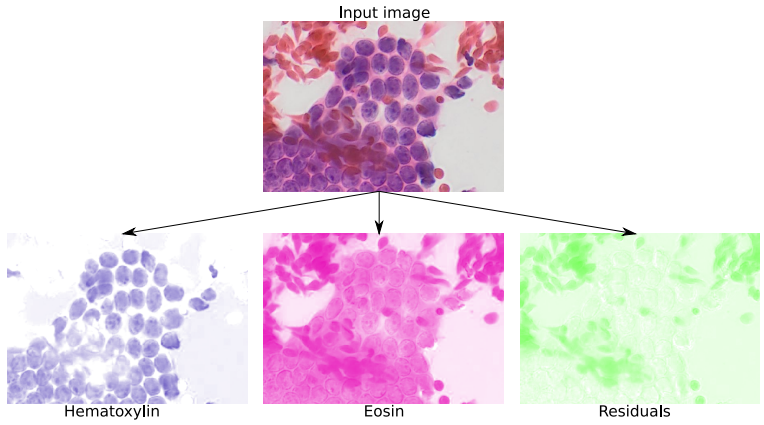


Figure 2.5: The effect of image deconvolution using the H&E method using Fiji software

The misclassification of erythrocytes as cell nuclei, especially in the case of images without cell nuclei, is the most serious drawback of the Otsu method for processing biopsy images. Unfortunately, such images occur during the cropping of larger images.

2.4 Summary

As a result of the described procedures, 2 training sets and 2 test sets were obtained. The first training set consists of images from the BreakHis collection from the laboratory in Parana and fibroadenomas from the University Hospital in Zielona Gora. Models created on the basis of this training set will be tested on data without fibroadenomas from the Hospital in Zielona Gora. The second set of training data consists only of data from the Hospital in Zielona Gora, which will be tested on two sets of training data. One comes from the BreakHis set combined with fibroadenomas cases from Zielona Gora, the second test data set contains only data from Brazil. The purpose of combining BreakHis data with fibroadenomas was to investigate the impact of data leakage between centers on the results of the best classification methods.

The images used for segmentation are of a different size from the those used for classification, i.e. the former are 700×460 pixels, which is a consequence of the size of images in the BreakHis set, and the latter are 230×230 pixels, which is a consequence of their use for deep neural networks. Therefore, the difference is dictated by the requirement of using different deep learning architectures.

The use of a fast pre-processing method allowed us to save image processing time while the main segmentation algorithm was running. The combination of H&E deconvolution with Otsu binarization is sufficiently effective in detecting diagnostically relevant images, but it proves insufficient for the segmentation of cell nuclei, in which case, an advanced segmentation method described in the next chapter must be applied.

Chapter 3

IMAGE NORMALIZATION BY SEGMENTATION

3.1 Introduction

Image segmentation is one of basic image processing techniques. Its role is to divide images into homogeneous components. In the case of cytological images, image segmentation focuses on separating cell nuclei from their background as well as on separating cell nuclei from each other. This means that in addition to point segmentation methods, it seems necessary to use edge detection methods. Combining these two tasks may be crucial in the process of segmenting cytological images. There are various approaches related to image segmentation, involving morphological transformations or the use of CNN networks. The need for manual selection of parameters and lower effectiveness make approaches based on morphological transformations less efficient than methods based on artificial neural networks. However, the CNN-based approach requires more time to prepare data and also requires a large amount of data in the training set.

3.2 Segmentation using a convolutional neural network

Neural networks are used in the task of detecting objects in an image and segmenting them. Three similar processes can be distinguished within these methods, i.e. object detection, semantic segmentation and instance segmentation. Object detection involves detecting objects and their position within the image, usually with the help of a bounding box, whereas semantic segmentation involves separating objects belonging to the same class at the pixel level. In the case of the performed experiments, there are three classes of objects that are subject to segmentation: the interior of cell nuclei, the edge of the nuclei and the background. Instance segmentation involves separating individual objects from a selected class of objects. In the conducted research, the overall segmentation algorithm consists of two main stages: semantic segmentation and instance segmentation. The differences between the basic concepts defined in the above paragraph are illustrated in Fig. 3.1.

In this work, object detection will not be used, so semantic segmentation, or more precisely, the U-Net CNN network, which implements this segmentation, will be the first method discussed in more detail. To process images of arbitrary sizes, an image overlapping strategy was used to ensure smooth segmentation, as described in the original approach used in the publication by Ronneber-

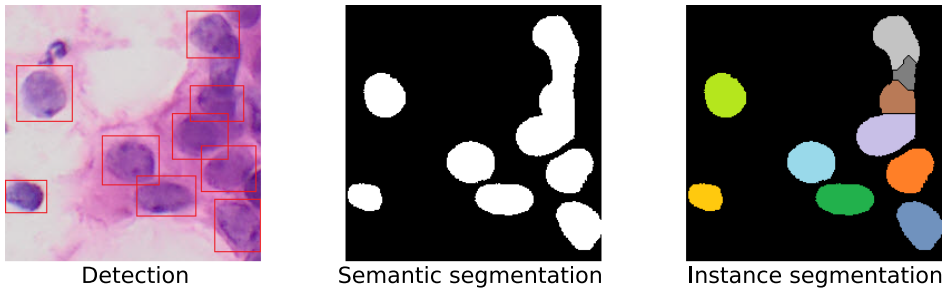


Figure 3.1: Differences between detection, semantic segmentation and instance segmentation

ger (Ronneberger *et al.* 2015). It assumes that an input image of size $572 \times 572 \times 1$ is transformed into an image of size $388 \times 388 \times 2$. This means that as a result of repeated convolution operations, the dimensions of the output image are significantly reduced, whereas only the central part of the image is subject to segmentation. The boundary problem can be solved using padding. Additionally, it can be noted that the input image is single-channel, but this does not exclude the possibility of using a three-channel input image. Further analysis of the U-Net approach will be supplemented with a diagram (Fig. 3.2) of the network modeled on the original drawing from the article (Ronneberger *et al.* 2015).

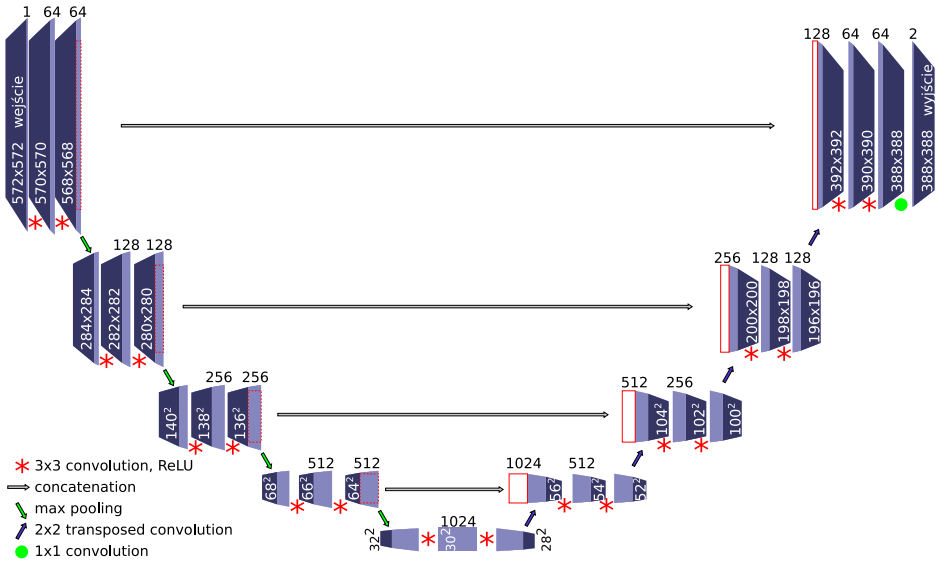


Figure 3.2: U-Net network diagram based on the article by Ronneberger

The diagram shows a characteristic shape of the network resembling the letter U and clearly two main parts of the network in the form of its encoding and decoding parts. The input image visible on the leftmost part of the diagram is processed using convolution and a $3 \times 3 \times 64$ filter. Performing two such operations changes the dimensions of the tensor from $572 \times 572 \times 1$ to $568 \times 568 \times 64$. The two convolution operations conclude with a maximizing pooling layer that reduces the width and height dimensions by half. Further layers are similarly subjected to two convolution operations, followed by a maximizing connecting layer, and this scheme is repeated three more times. After its completion, we obtain a $32 \times 32 \times 512$ tensor, which is subjected to two further convolution operations, whose outcome is a $28 \times 28 \times 1024$ tensor. It is from this tensor that image reconstruction begins using a transposed convolution layer that is used for expansion (upsampling). As a result of transposed convolution, a $28 \times 28 \times 1024$ tensor is transformed into a $56 \times 56 \times 512$ tensor. Additionally, the obtained tensor is at this point connected to a section of the corresponding layer from the set of layers in the first part of the network. This procedure is intended to supplement the second part of the network with information from the first one in order to improve the segmentation results. Subsequently, the obtained tensor is subjected to two convolution operations. Starting with transposed convolution, these activities are repeated three more times until a $392 \times 392 \times 64$ tensor is obtained, which is supplemented with a slice of the same dimensions obtained from the first layer after two convolutions. The resulting tensor is subjected to a double convolution operation ($3 \times 3 \times 64$) followed by a final convolution with a $1 \times 1 \times 1$ filter. The last convolution reduces the tensor to the resulting image, which contains objects in two classes.

3.3 Hybrid segmentation system

Watershed segmentation is an algorithm inspired by topographic characteristics of land depressions and their flooding with water. In the literature related to this area, it is possible to find a definition of three basic groups of points in an image, i.e.: (I) local minima, (II) points after placing the water in which it will tend to one of the minima and (III) points at which water will tend to at least two minima with equal probability. The points in group III are called watersheds, hence the name of the algorithm. In it, the input image is typically given a binary representation (e.g., cell nuclei area and background). Such a binary image can be obtained as a result of thresholding or during semantic segmentation by means of a CNN network. After the areas of cell nuclei have received the value 1 and the background 0, the image can be treated as a binary mask. In the next step, the surface of binary images undergoes Euclidean transformation in such a way that the pixels receive a value indicating the distance of the examined pixel from the edge of the objects. The obtained values can be imagined as depths relative to the surface (background). The points furthest from the edge of the objects become local minima, from which the process of flooding the pools begins. In the last step of the algorithm, areas are flooded, starting from the minima, until

the areas assigned to different minima meet in the watershed. Unfortunately, in this form the algorithm tends to create an excess set of segments (Gonzalez & Woods 2017, Yang, Li & Zhou 2006). The main cause of over-segmentation is thought to be rooted in a number of potential minima (Gonzalez & Woods 2017). To deal with this problem, a modified version of the watershed algorithm can be used, which uses additional markers marking the center points of cell nuclei. Even though detecting kernel midpoints is challenging, the ensuing improvement in the quality of semantic segmentation is of key importance. The modified approach is labeled as marker-controlled watershed algorithm. It differs from the classical approach in that in the modified version the starting points from which the flooding of the topographic surface begins are not local minima but designated markers.

To deal with the presented problem, a hybrid segmentation method was proposed (Fig. 3.3), which made use of two U-Nets and the watershed method with markers. The first network was trained to predict whether pixels belonged to the interior of the nucleus (class 1) or to the edges of the nucleus and the background (class 2). Based on the semantic segmentation provided by this network, the interior of the nuclei was extracted in the form of binary images. These images were used to determine the topographic surface for the watershed algorithm using the distance transform. The second U-Net was trained to detect nuclei centers. The network predicted whether pixels belonged to the center of the nucleus (class 1) or to the interior of the nucleus, edges and background (class 2). Centers (nuclei markers) were extracted from the results of semantic segmentation in the form of binary images.

In the last step, the topographic surface was connected to centers (nuclear markers) using morphological reconstruction. The modified topographic surface was processed by classical watershed transformation to detect nuclei.

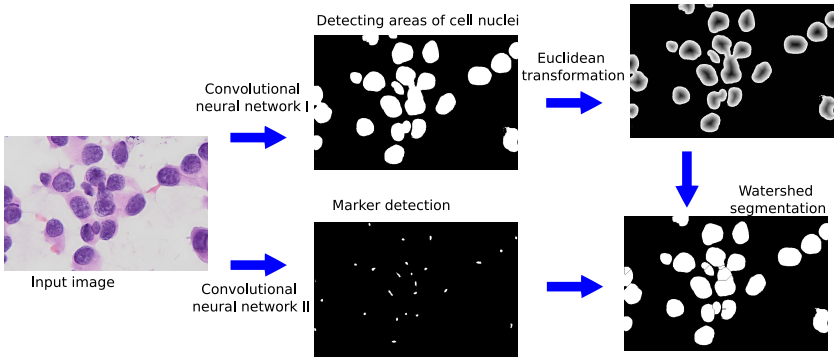


Figure 3.3: Hybrid method of watershed segmentation using initial segmentation using a CNN network and detecting watershed markers using a second CNN network

3.4 Evaluation methods

Appropriate metrics should be used to assess the quality of segmentation. Evaluation of the results begins by marking cell nuclei in the test image. This activity most often comes down to precisely marking the edges of the cell nucleus using a pointing device on a computer. Nuclei can be selected by means of ROI (Region Of Interest) objects and this is a standard technique for selecting regions in medical images. In practice, the ROI object is the envelope of the cell nucleus, which can then be transformed into a binary object mask or a labeled mask (Fig. 3.4).

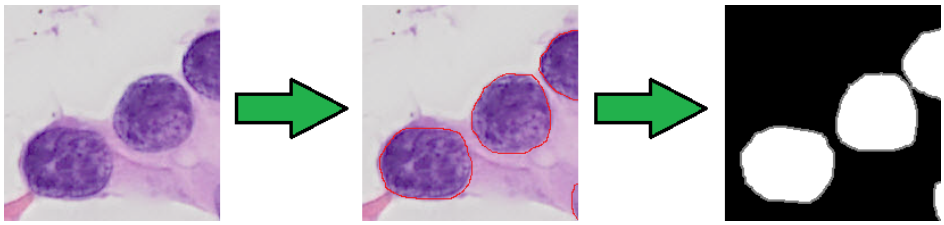


Figure 3.4: Stages of transition from the real object to the ROI and binary mask

Jaccard index

The binary mask is useful for one of the most intuitive metrics called the Jaccard Index (JI) (Jaccard 1912). This metric is based on set algebra and measures the similarity between two sets:

$$JI(X,Y) = \frac{X \cap Y}{X \cup Y} \tag{3.1}$$

The Jaccard index is calculated using cell nuclei masks obtained from segmentation with masks derived from manually marked ROIs. The method is effective because binary masks are collections of pixels. From a practical point of view, it is important to determine the size of the bounding box in which the mask of the largest cell nucleus can fit. This is important because the algorithm processes single rectangles inside which there is a cell nucleus. The size of the rectangle therefore depends on the area of the largest object and affects the speed of the calculation processing. The criteria for determining a central point of a binary mask is another issue that affects the obtained result. Depending on the adopted method that determines the center, a slight shift of the central point may slightly change the final result of the Jaccard Index. The JI value ranges from 0 to 1, where the higher the value, the better the objects match each other. Sometimes, instead of the Jaccard Index, you can use the Jaccard Distance, which is the complement of the Jaccard Index to 1:

$$JD(X,Y) = 1 - \frac{|X \cap Y|}{|X \cup Y|} \tag{3.2}$$

The advantage of JD is that it obtains direct information about the value of the similarity error between the two compared object masks.

Dice-Sørensen coefficient

Dice-Sørensen Coefficient (DSF), also known by other names: F-score, Czekanowski Index, Steinhaus Index or Sørensen Index, is another measure often used in the process of evaluating segmentation results (Dice 1945, Sørensen 1948). This quantity is also based on the algebra of sets and assumes the following form:

$$WDS(X,Y) = \frac{2|X \cap Y|}{|X| + |Y|} \tag{3.3}$$

As in the case of JI, calculation by means of the Dice-Sørensen coefficient requires binary masks of objects or their coordinates. It also takes values from 0 to 1, where 1 means that the objects are identical. Due to their mathematical similarity, the Jaccard Index and the Dice-Sørensen Index are often used interchangeably.

Hausdorff distance

There is a measure completely different in its assumptions from JI and DSF, i.e. the distance or Hausdorff metric (DH). It is often used to assess the effectiveness of segmentation of cell nuclei. It measures the distance between two sets in metric space (Fig. 3.5). The mathematical notation is defined as follows:

$$DH(A,B) = \max\left\{\sup_{a\in A}\inf_{b\in B} d(a,b), \sup_{b\in B}\inf_{a\in A} d(a,B)\right\}, \tag{3.4}$$

where sup stands for supremum and inf stands for infimum.
In the case of cell nuclei, we deal with data in two-dimensional space. The nucleusof

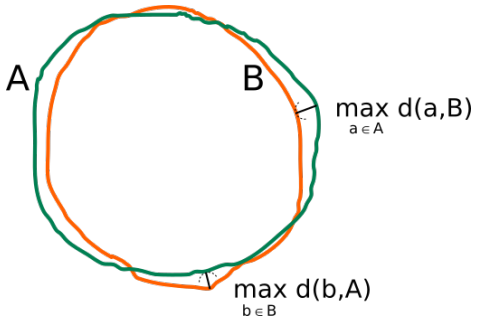


Figure 3.5: Graphical interpretation of DH

a single cell is a set of pixels: $A = \{a_1, a_2, a_3, ..., a_m\}$ in the image space. The Euclidean distance between individual elements of set A and set $B = \{b_1, b_2, b_3, ..., b_n\}$ in the two-dimensional Euclidean plane can be defined as:

$$d_E(a,b) = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2} \tag{3.5}$$

We can then define the distances between each pixel and another set of pixels:

$$d(a,B) = \min_{b\in B} d_E(a,b); \qquad d(b,A) = \min_{a\in A} d_E(b,a) \tag{3.6}$$

Finally, we can calculate DH as (Fig. 3.5):

$$d_H(A,B) = \max\{\max_{a\in A} d(a,B), \max_{b\in B} d(b,A)\} \tag{3.7}$$

However, when processing binary kernel masks, DH is limited to examining the edges between masks. Using this approach can significantly speed up the DH calculation. Cell nuclei masks turn DH units into pixels. DH should be interpreted in such a way that the higher the result, the lower the match between the two sets.

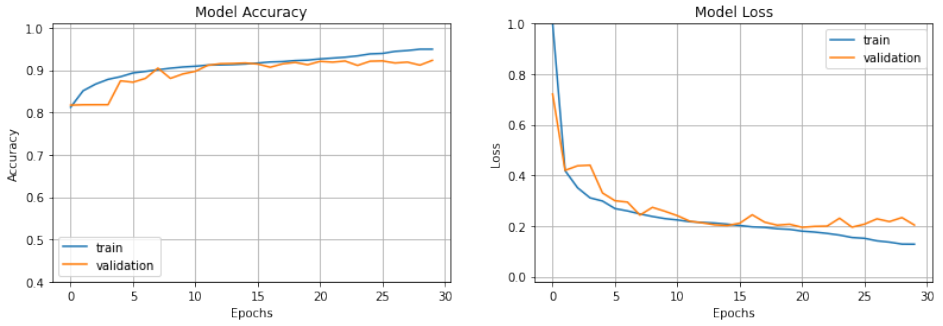
3.5 Verification of the accuracy of the hybrid segmentation method

3.5.1 Implementation of a hybrid segmentation method

The results obtained in the publication by (Kowal *et al.* 2018) made it possible to establish an effective method of segmenting cell nuclei based on a U-Net CNN network supported by a marker-controlled watershed algorithm. For conclusive separation of individual cell nuclei, additional processing of images segmented with an artificial neural network is necessary. This is due to the fact that neural networks generate a probability result that does not define the edges between the clustered objects. In other words, the result depends on the initial separation of cell nuclei in the images. The network used for segmentation was taken from the Keras documentation (*https://keras.io/*) (Chollet 2017a), which in turn is an interpretation of the U-Net (Ronneberger *et al.* 2015) network. The only difference is the size of the input images (700 × 460) and the additional bonus of the U-Net segmentation with the watershed algorithm. The constructed model of an artificial U-Net network is presented in a descriptive form in the Appendix C.

The analysis of the network structure revealed that the input 700 × 460 image was transformed to the size of 704 × 464. This procedure allowed the subsequent stages of training to be completed without deforming the shape of the output image. In turn, the tensor of the smallest image has dimensions of 88 × 58 × 256, which means that in the lowest layer 256 88 × 58 images obtained from a three-channel 704 × 464 image were modeled. The total number of model parameters was 2 058 979, of which 2 055 203 were subjected to learning. The model uses the gradient steepest descent algorithm RMSprop. The network is not particularly demanding when it comes to hardware resources (see: Appendix B), and the final training result can be obtained after only several dozen minutes.

Sample modeling results are presented in Fig. 3.6. Both the loss plot and the accuracy plot reveal that the learning curve for the validation data terminates improvement approximately in the 20th training epoch. However, the parameters of the curves for the training data are improving, which means that the model is starting to overfit. Detailed data indicate that the smallest model loss for validation data occurred in the 21st training epoch and amounted to 0.1947, while the model accuracy in this epoch was 0.9207. For training data in the 21st epoch, the loss value was 0.1796 and the accuracy was 0.9264. Therefore, epoch 21 was the optimum that could be obtained from these data.



(a) Accuracy of the model in subsequent epochs **(b)** Loss of the model in subsequent epochs

Figure 3.6: The course of training the U-Net model

Another important aspect was the preparation of training and test data. The training dataset contained 844 manually segmented 700 × 460 images. Segmentation was performed by means of ImageJ software (Fiji) using the ROI file creation tool. The marking of cell nuclei was performed using a computer mouse and partly using a graphic tablet, which turned out to be a more convenient tool and significantly accelerated the process of marking images. The total number of manually segmented images amounted to 844, including: 89 benign images (from 3 patients) and 62 malignant images (from two patients), all from the BreakHis collection; 29 images (from 1 patient) from the GZG collection, as well as 330 benign images (from 25 patients) and 334 malignant images (from 25 patients) from the SzUZG collection. The next stage of image preparation was to create a mask consisting of the following 3 categories: the interior of cell nuclei, the edge of the nucleus and the background. The first step in data preparation was to load the ROI file and create binary masks for each cell nucleus. While generating a single nucleus mask, the edges of the cell nucleus were also determined. To create the edges, a mask of the resulting cell nucleus was used, on which double morphological erosion with variable structural elements (Formula 3.8) and a single dilation were performed.

$$\begin{bmatrix} 1,1,1 \\ 1,1,1 \\ 1,1,1 \end{bmatrix} \quad \begin{bmatrix} 0,1,0 \\ 1,1,1 \\ 0,1,0 \end{bmatrix} \tag{3.8}$$

The envelopes created after erosion and dilation were combined into one object in the form of the edge of the cell nucleus (Fig. 3.7). Therefore, the test set included a total of 5288 images. Some patients from the training set overlapped with testing set, especially in the case of data from the SzUZG set. Due to the need to have all images for subsequent classification as well as to prevent leakage of test data to the training set, the cross-validation method was used in the Leave- One-Out. In other words, in order to obtain test data for a specific patient, images belonging to him or her were excluded from the training set. The segmentation experiment was therefore repeated a total of 55 times, and the entire duration of the series of

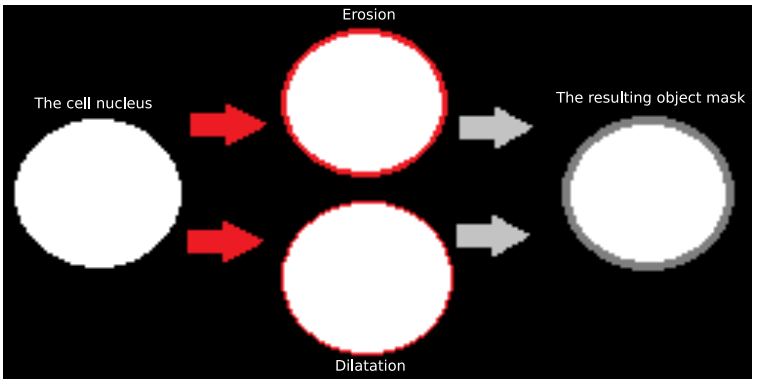
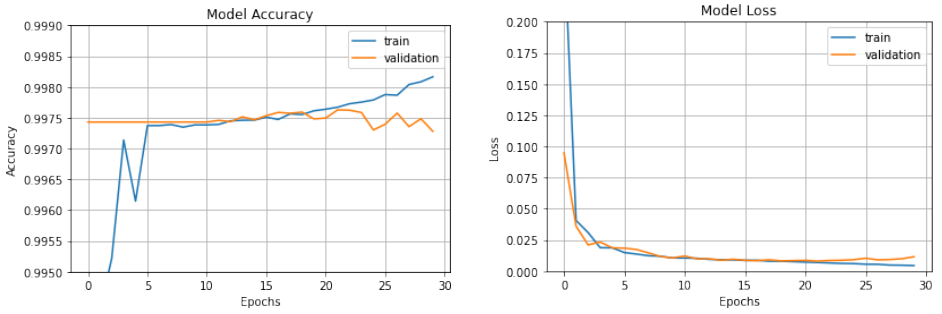


Figure 3.7: Scheme of creating edges of cell nuclei of images in the training set

experiments amounted to several weeks. Finally, all images were segmented using a CNN network. Interestingly, in all 55 experiments, comparable accuracy (0.92 ± 0.01) and loss (0.19 ± 0.01) results were obtained, and these results also occurred around the 20th training epoch. The end result of the CNN network operation consisted channels containing the probabilities of pixels belonging to particular classes (kernel interior class, edge class and background class). The images were processed in such a way that edges were excluded from the final map of cell nuclei. In other words, subtracting the edge class from the interiors of the nuclei enhanced the final result of separating single cell nuclei.

The final duration of the experiments was significantly influenced by the interleaving of training the network for segmenting cell nuclei with training the network for detecting cell nuclei centers. This experiment involved the detection of central points of cell nuclei. The points had an area of several pixels and their intended purpose was to be used as starting points (markers) for the watershed segmentation algorithm. Point detection is quite important because it allows for estimating the number of individual objects in an image, and thus detecting individual cell nuclei. The structure of the network is almost identical to that used for segmentation of cell nuclei, differing only in the structure of the resulting image, which consists of two channels (background and centers of cell nuclei). Just like in the first U-Net, the softmax activation function was used in the last layer. This is possible because in both models the resulting number of channels is greater than 1.

Examples of results of U-Net training used to detect central points are presented in Fig. 3.8. It is clearly visible that a very high accuracy value and a very small loss are achieved in initial training epochs. Only at higher magnifications can it be seen that the network achieves optimal results and then overfits in the vicinity of the 20th training epoch. The extremely high accuracy results and the rapid achievement of very low loss are dictated by the characteristics of the input data. The images entered into the network have a significant disproportion in the number of pixels belonging to particular classes (the class of the cell nucleus center and the background class). Hence, a conclusion can be drawn that neural networks



(a) Accuracy of the model in subsequent epochs **(b)** Loss of the model in subsequent epochs

Figure 3.8: Training flow of the U-Net model for central points

will not make a large error if they classify pixels that belong to the centers of cell nuclei into the numerically dominant background class. In such a case, there are several potential methods to prevent such a situation, e.g. introducing a penalty in the network for classifying pixels belonging to the centers as background. On the other hand, the problem may also be solved by processing the obtained images using thresholding. The use of binarization is possible because the result obtained at the output of the neural network represents in this case the probability of a pixel belonging to a particular class. Hence, it is very easy to experimentally select the binarization threshold, which was finally set at 0.10.

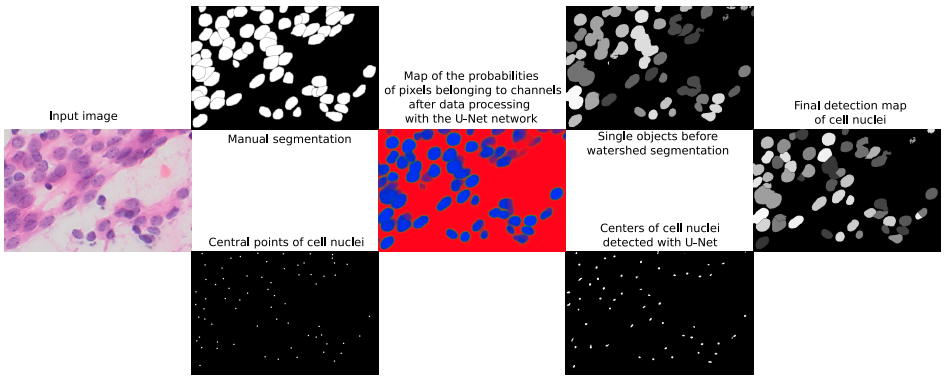


Figure 3.9: Scheme of segmentation of cell nuclei

Additionally, Fig. 3.9 shows a general image processing scheme that was performed in the experiment. On the right side, the final effect of image processing after extracting individual cell nuclei is visible. The upper part of the image presents how the neural network successfully extracted objects belonging to the class of cell nuclei, but was unable to disconnect the existing objects, as this task requires an additional post-processing method.

3.5.2 Results of segmentation of cell nuclei

The obtained results largely depend on the quality of the segmented image. There is a noticeable tendency to decrease segmentation accuracy for images in which cell nuclei have not absorbed a sufficient amount of hematoxylin. On the other hand, the method avoids the detection of red blood cells as cell nuclei and is exceptionally effective in detecting clusters of clumped cell nuclei. Sample results are presented in Fig. 3.10. The results for a benign case from the SzUZG set reveal that the network detects cell nuclei in places where difficulties occurred even during manual segmentation.

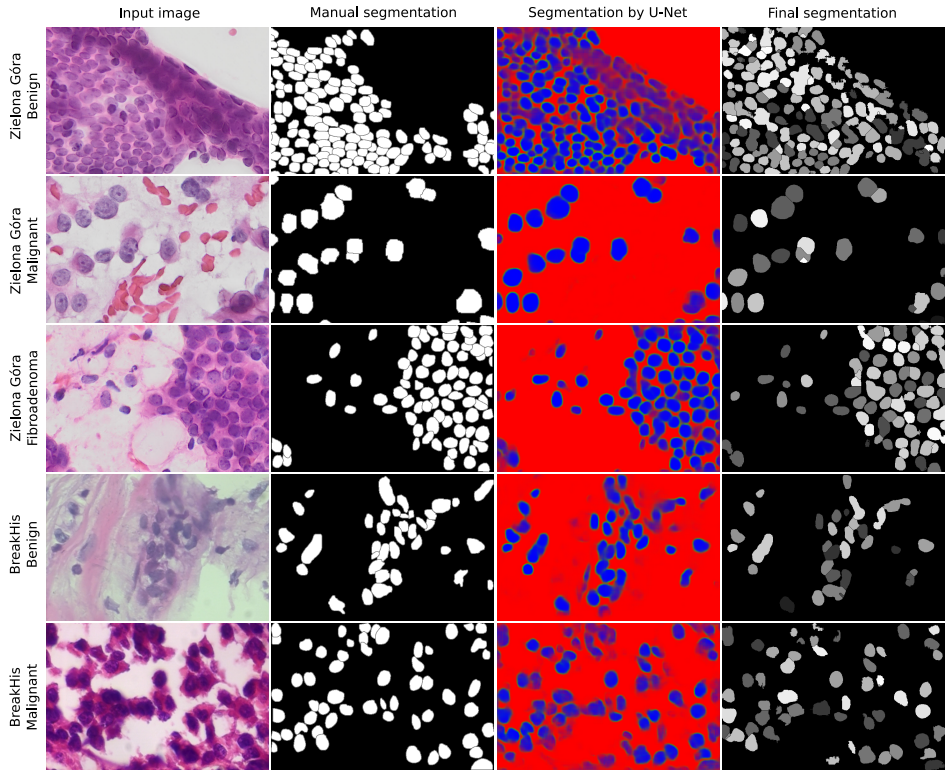


Figure 3.10: Example of segmentation results

Further results include quantitative evaluations that numerically characterize segmentation accuracy. The quantitative evaluation method used in the experiment consists of several stages. The first stage compares individual cell nuclei from manually marked images with cell nuclei detected by means of a neural network. During this stage, the value of JI was examined. If the index value exceeded 0.5, i.e. if at least 50% of the pixels of the detected cell nucleus coincided with the pixels of the cell nucleus from the manually marked image, then such a nucleus was included in the Detected group. If the cell nucleus in the manually marked image

had no counterpart (below a value of 0.5) in the image segmented with a neural network, it was included in the Undetected set. The third group of objects were labelled as Over-segmented, i.e. cell nuclei detected by the neural network, which did not find their counterparts in manually segmented images. JI and DH values were obtained for all detected objects. Additionally, the percentage values of Detected, Undetected and Over-segmented nuclei were calculated:

$$D[\%] = \frac{D}{D+U}100 \tag{3.9}$$

$$U[\%] = \frac{U}{D+U}100 \tag{3.10}$$

$$O[\%] = \frac{O}{D+O}100, \tag{3.11}$$

where D mean Detected, U mean Undetected and O mean Over-segmented.

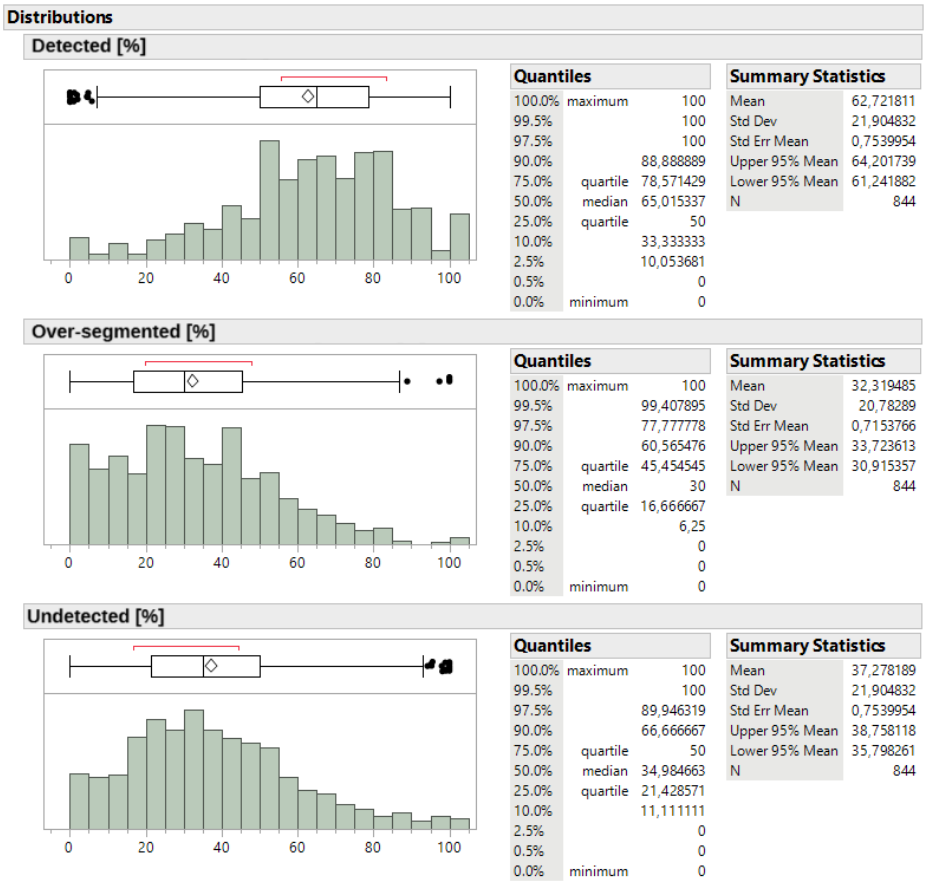


Figure 3.11: Value distributions and statistics

The percentage of detected cell nuclei, i.e. having their counterpart in the manually segmented set, was the first analyzed key feature. The average percentage of correctly detected cell nuclei is 62.7%, while the median is 65.0%. Additionally, it is observed that the distribution of detected cell nuclei is strongly asymetric to the left, which means that high detection values dominate. The distributions of undetected and over-segmented objects are asymmetric on the right, which in turn denotes the dominance of low values. The average percentage of undetected cell nuclei stood at 37.3 and of over-segmented at 32.3, whereas the medians amounted to 35.0 and 30.0, respectively. These values are much lower than the average value of detected cell nuclei. The empirical distribution of detected cell nuclei also reveals a small number of disturbing results, i.e. those where the number of detected objects failed to exceed 20%. In the entire set of 844 observations, there were 36 such cases, which means that the neural network had a particular problem with segmenting 4.3% of the images. Analyzing a random image that ended up in the low segmentation group of images should clarify some of the potential reasons for the poorer results. The example in Fig. 3.12 presents an image whose evaluation

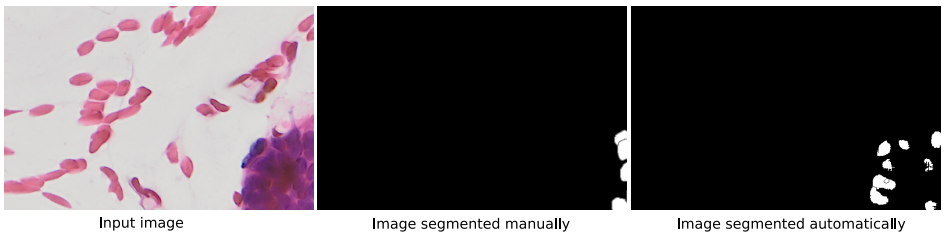


Figure 3.12: Example image with a low segmentation evaluation score

result was set at 0% detected, 100% undetected and 100% over-segmented. The input image in this example is characterized by a very tightly clumped structure of cell nuclei. Additionally, the area occupied by cell nuclei is saturated and slightly covered with acidophilic structures (eosin) and erythrocytes. The difficulty in manually assessing the edges of cell nuclei is reflected in the central part of the image, where only 4 cell nuclei are marked. However, the neural network managed to detect areas of cell nuclei with clear separation and good saturation with hematoxylin. Unfortunately, the detected cell nuclei were not included in the manually marked set. Therefore, the next conclusion concerns the method itself and the accuracy of determination, which decreases with the increase in the complexity of cellular structures. It is also an example of discrepancies between human and computer visions, as well as the imperfections of quantitative evaluation methods and the need to supplement them with qualitative evaluation methods.

A randomly selected image (Fig. 3.13) with an average percentage of correctly detected cell nuclei (58.3%) provides the second example. In it, Over-segmented cell nuclei constitute 18.1%, while those Undetected amount to the level of 41.3%. The obtained quantitative results do not reflect the actual quality of segmentation and considering the visual qualitative effect of the obtained result, this example should be seen as a very good segmentation result.

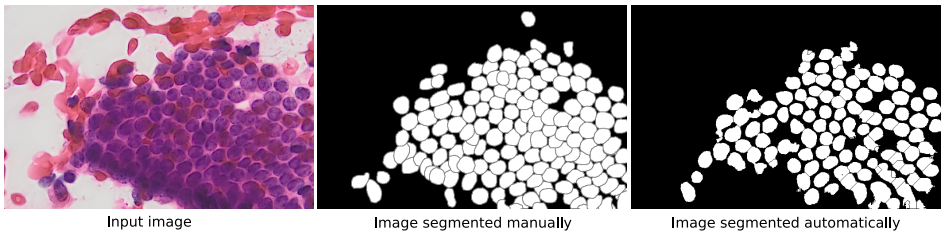


Figure 3.13: An example image with an average segmentation evaluation result

Further analysis of the results begins with the presentation of distributions for the categories into which the sets can be divided (Fig. 3.14). The first cat-

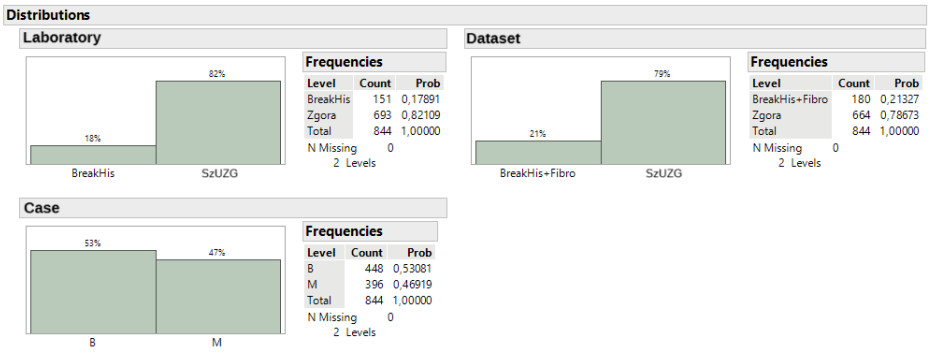


Figure 3.14: Distributions of the number of features in groups

egory (Laboratory) refers to the medical facility in which the sample originated. The second category (Dataset) refers to the data set prepared in the study, i.e. BreakHis + GZG data, where the second set includes malignant and benign cases with GZG. The third category (Case) points to the type of cancer in the images, i.e. benign (B) or malignant (M).Large disproportion in the number of marked images between centers is the first noticeable problem which may affect the results of the analyses. The ratio of the number of malignant cases (47%) to benign cases (53%) is much better. It would also be interesting to analyze whether the quality of segmentation depended on the centers in which the images originated and whether the types of cancer affected the accuracy of segmentation. The analysis of the impact of the images' origin begins with the analysis of box plots (Fig. 3.15).

The graphs in the left column illustrate an advantage in the detection efficiency of images from the Hospital in Zielona Góra (SzUZG). All statistical parameters are higher for the group of images from Zielona Góra, for which the maximum line reaches 100%. For comparison, the maximum for the BreakHis set stops at 75%, while one observation of 100% is treated as an outlier. The over-segmentation plots in the middle column show slightly different conclusions. The average value of the percentage of over-segmented cell nuclei is lower for the

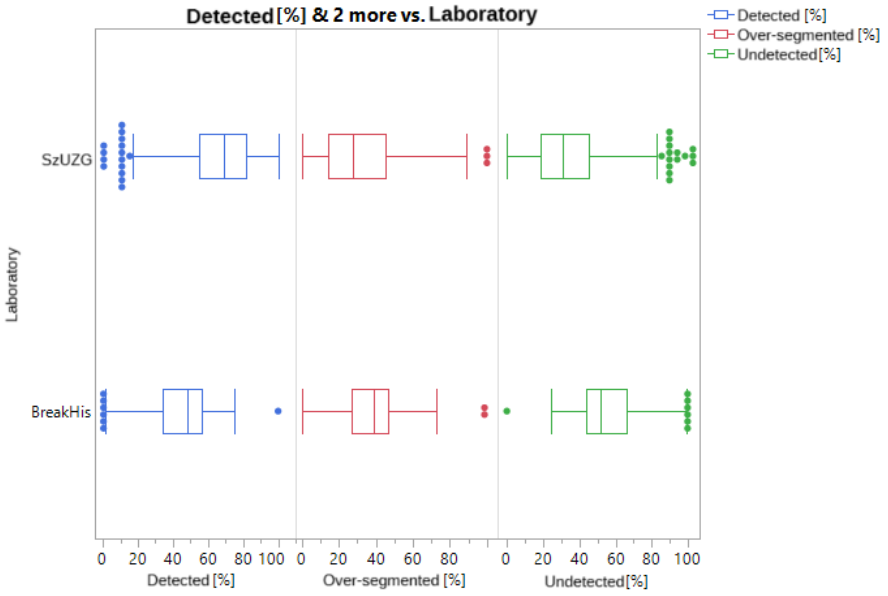


Figure 3.15: Boxplots of segmentation results to the center of origin of the samples

Zielona Góra collection than for the BreakHis collection. On the other hand, the spread of the percentage of the number of over-segmented objects is lower for the BreakHis set. Moreover, the graph for the BreakHis set shows greater symmetry. The last group of graphs (for undetected objects) is a mirror image of the graphs for detected objects, so the conclusions here are a mirror reflection of conclusions drawn for the first column.

After analyzing the box plots, the first observations come to mind regarding the impact that individual medical laboratories have on the detection of cell nuclei. Further analyzes will concern the curves of matching the image acquisition center to the percentage of detected, over-segmented and undetected objects (Fig. 3.16).

The graph on the left shows the logistic curve of matching the percentage value of detected objects to the individual centers from which the data originates. The chart shows that as the percentage of the number of detected nuclei increases together with the percentage of images from the Hospital in Zielona Góra. The chi-square test statistic in this model tests a hypothesis that the factor (detected [%]) has no effect on the response variable (this variable was labeled as Osrodek). The P-value next to the test statistic was estimated at < 0.001, which means that there is a strong justification for rejecting the null hypothesis and accepting the hypothesis that the result of cell nuclei detection is dependent on the image acquisition center as correct. The situation is similar for the graph on the right, with the difference that an increase in the number of undetected objects causes an increase in the percentage of images from the BreakHis set. Taking into account the p-value, also in this case the hypothesis about the lack of influence of the

percentage of undetected objects relative to the center of origin should be rejected. The middle graph shows a slightly smaller slope, so it can be concluded that the influence of these features relative to each other is slightly lower than in the previous cases. However, there is still a noticeable tendency for the percentage of images from the Zielona Góra collection to decrease as the percentage of over-segmented objects increases. The null hypothesis, which states that there is no relationship between the number of over-segmented objects and the center of origin of the images, should be rejected.

Moving on to the next analysis, it is worth recalling that the rest of the experiment will be based on two sets. The first set contains images from the SzUZG set, while the second contains BreakHis + GZG images. The GZG set was combined with the BreakHis set to balance the number of benign and malignant cases. Finally, there is a brief analysis of the impact of the tested set on the efficiency of cell nuclei segmentation (Fig. 3.17). Unfortunately, the obtained results are similar to those obtained based on the center of origin. Strong impact of the dataset on detection and over-segmentation is still visible. In other words, images from the BreakHis + GZG set will most likely be worse segmented than the images from the SzUZG set. On the one hand, this may be due to the fact that each patient from the Zielona Góra collection had at least several images marked. In the BreakHis set, this only applies to several patients. Despite this discrepancy, the overall segmentation results of the BreakHis set are good, which in turn proves that the U-Net network performs excellently not only on data from one research center, but also has excellent generalization properties.

The next analysis involves verifying the influence of the type of cancer in relation to detection. In the Fig. 3.18, the graph for Detected and Undetected is almost flat. This may mean that these features have little impact on the type of cancer in the image. To confirm the visual observation, it is possible to check the p-value regarding the hypothesis which states that the (detected [%] and undetected [%]) factors have no effect on the response variable (Chance). The P-value for both factors is 0.6524, which means that we have no strong grounds to reject the null hypothesis. Therefore, we can conclude that these factors have no effect on the type of cancer. In other words, object detection is similar in malignant and benign cases. The graph for over-segmented objects is slightly different. Here we can already see a tendency that there is a larger number of over-segmented objects for malignant cases. The obtained p-value is 0.0132, so it is lower than the adopted confidence level of 0.05, which means that we have grounds to reject the null hypothesis and accept an alternative hypothesis that the factor of the percentage of over-segmented objects influences the type of cancer in the image. In general, there is a much smaller impact of the type of cancer on the detection result.

The final analysis of the segmentation results will focus on matching the datasets and cancer cases to the evaluation metrics (JI, DH). The results shown in Fig. 3.19 suggest that the results of both evaluation metrics have an impact on both the set from which the image derives and the type of cancer. Generally, it can be seen that benign cases obtained better evaluation results than malignant

ones, although in the case of JI it is not so obvious, because with higher object matching values, malignant cases have an advantage in the evaluation. There are clearer evaluation results for the influence of the origin. In this case, better values are obtained by samples from the Zielona Góra dataset, both for the DH and JI metrics.

3.6 Summary

This chapter describes the process of segmenting cytological and histopathological images using a hybrid method based on CNN networks and a marker-driven watershed algorithm. A developed method which uses two CNN networks and a watershed algorithm with markers constituted the main segmentation tool. The first network performed image segmentation, while the second one performed the task of detecting the central points of cell nuclei. The data prepared for the study were collected from two different medical centers. Despite the small number of marked examples, the network coped with the segmentation problem very well, achieving results comparable to those in the literature (Lagree *et al.* 2021). A much shorter execution time of a single experiment is an advantage of the proposed approach over other approaches (Lagree *et al.* 2021). In order for the research conducted in the dissertation to be reliable, it required the use of Leave-One-Out cross-validation. This approach significantly increases the total duration of the experiment, but avoids a situation in which images from one patient are included in both the training and test data sets. The duration of the segmentation experiments amounted to several weeks in total. However, with the segmentation models from the article (Lagree *et al.* 2021), the duration of the study would be increased many times over.

Despite obtaining similar quantitative evaluation results in the proposed approach in comparison to those in the literature, it is difficult to fully compare the obtained results. This problem results from insufficient standard database and lack of insight into the division of databases used when creating training and test data. The data used also differs in quality and resolution. In general, the segmentation results obtained on the basis of the developed method are of high quality and this method has high generalization abilities.

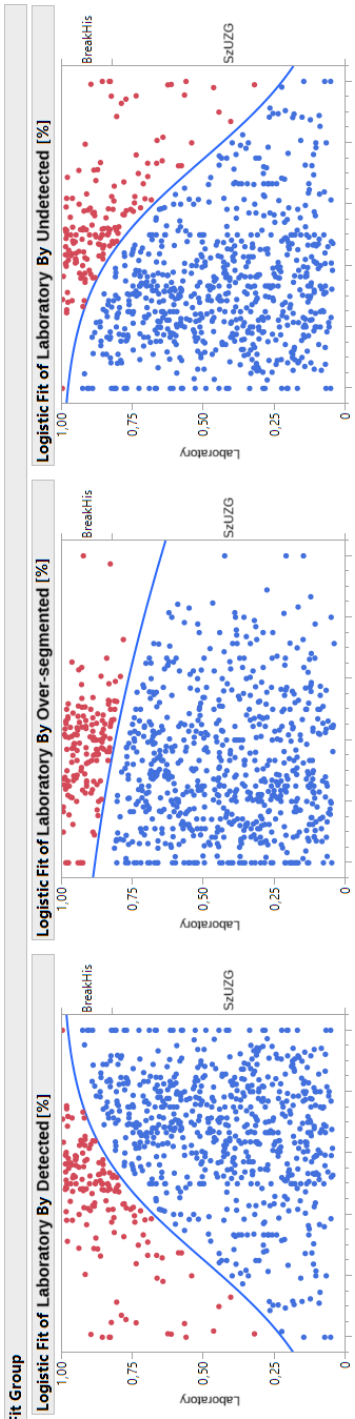


Figure 3.16: Matching segmentation results to the center of origin of the samples

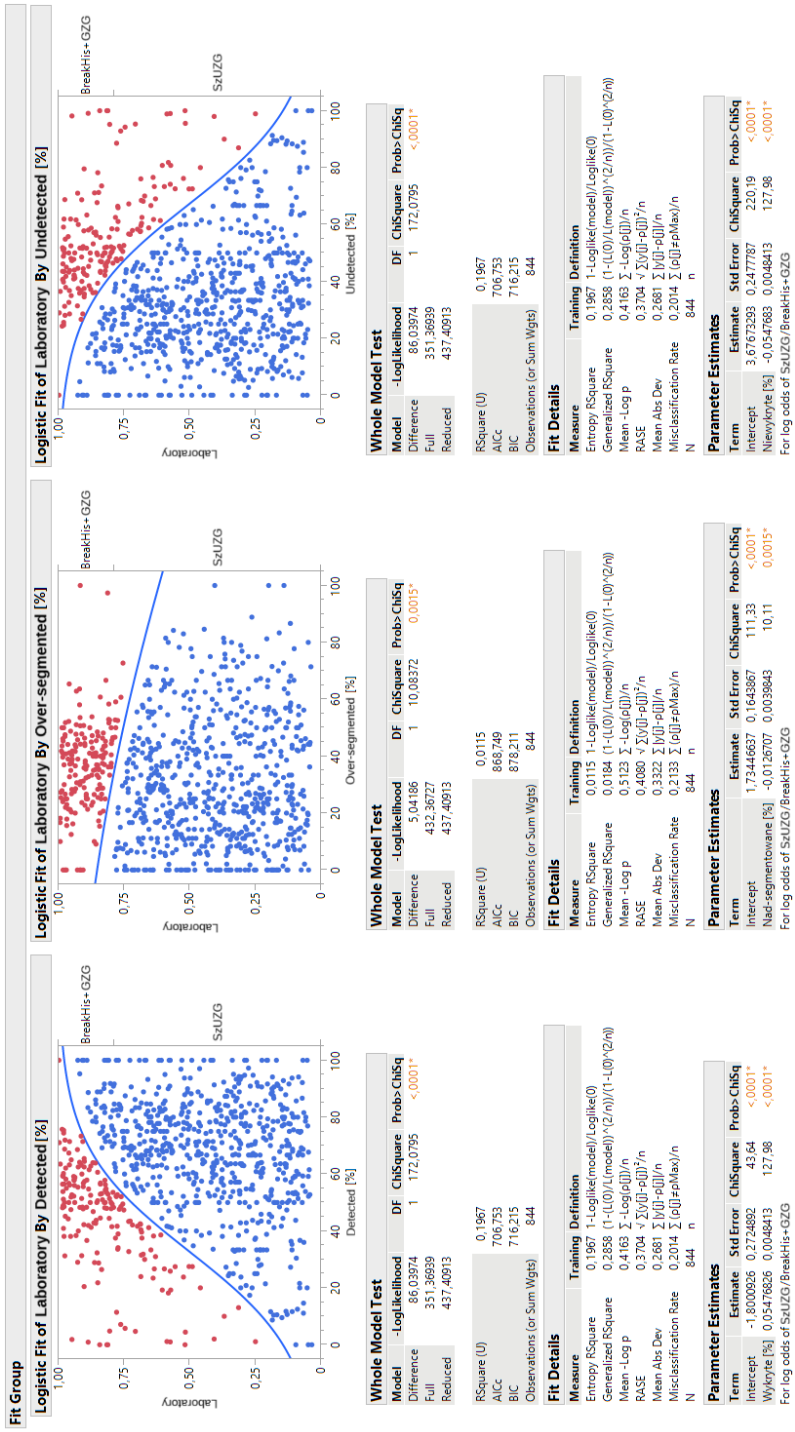
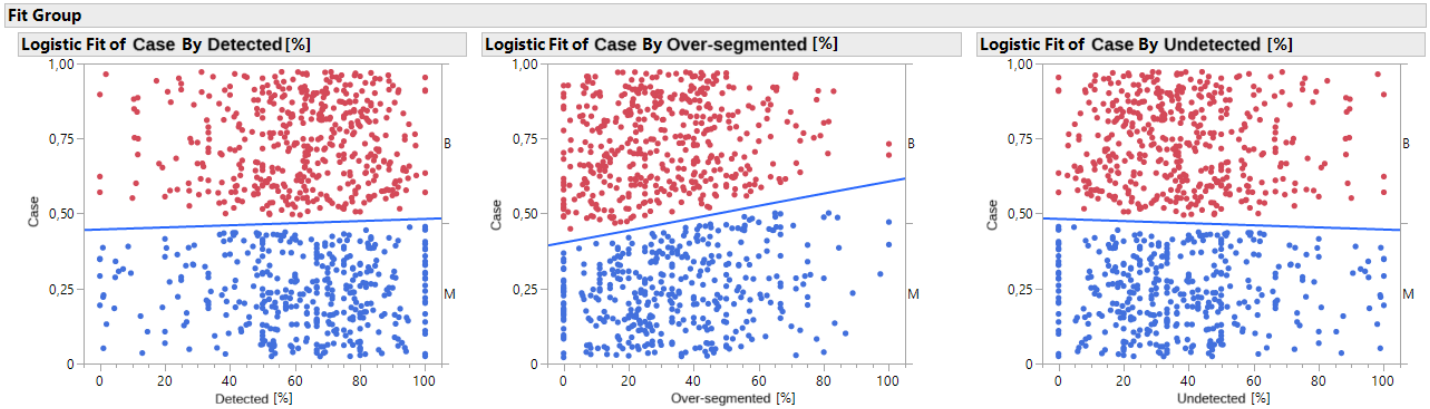


Figure 3.17: Matching segmentation results to the created sets



Whole Model Test

Model	-LogLikelihood	DF	ChiSquare	Prob>ChiSq
Difference	0,10142	1	0,202836	0,6524
Full	583,31189			
Reduced	583,41331			

RSquare (U)	0,0002
AICc	1170,64
BIC	1180,1
Observations (or Sum Wgts)	844

Fit Details

Measure	Training	Definition
Entropy RSquare	0,0002	1-Loglike(model)/Loglike(0)
Generalized RSquare	0,0003	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0,6911	∑ -Log(p[j])/n
RASE	0,4990	√ ∑(y[j]-p[j])²/n
Mean Abs Dev	0,4980	∑ |y[j]-p[j]|/n
Misclassification Rate	0,4692	∑ (p[j]≠pMax)/n
N	844	n

Parameter Estimates

Term	Estimate	Std Error	ChiSquare	Prob>ChiSq
Intercept	-0,2124624	0,2096231	1,03	0,3108
Wykryte [%]	0,00141979	0,0031537	0,20	0,6526

For log odds of M/B

Whole Model Test

Model	-LogLikelihood	DF	ChiSquare	Prob>ChiSq
Difference	3,07008	1	6,140159	0,0132*
Full	580,34323			
Reduced	583,41331			

RSquare (U)	0,0053
AICc	1164,7
BIC	1174,16
Observations (or Sum Wgts)	844

Fit Details

Measure	Training	Definition
Entropy RSquare	0,0053	1-Loglike(model)/Loglike(0)
Generalized RSquare	0,0097	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0,6876	∑ -Log(p[j])/n
RASE	0,4972	√ ∑(y[j]-p[j])²/n
Mean Abs Dev	0,4945	∑ |y[j]-p[j]|/n
Misclassification Rate	0,4419	∑ (p[j]≠pMax)/n
N	844	n

Parameter Estimates

Term	Estimate	Std Error	ChiSquare	Prob>ChiSq
Intercept	-0,3909686	0,1288422	9,21	0,0024*
Nad-segmentowane [%]	0,00825769	0,003347	6,09	0,0136*

For log odds of M/B

Whole Model Test

Model	-LogLikelihood	DF	ChiSquare	Prob>ChiSq
Difference	0,10142	1	0,202836	0,6524
Full	583,31189			
Reduced	583,41331			

RSquare (U)	0,0002
AICc	1170,64
BIC	1180,1
Observations (or Sum Wgts)	844

Fit Details

Measure	Training	Definition
Entropy RSquare	0,0002	1-Loglike(model)/Loglike(0)
Generalized RSquare	0,0003	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0,6911	∑ -Log(p[j])/n
RASE	0,4990	√ ∑(y[j]-p[j])²/n
Mean Abs Dev	0,4980	∑ |y[j]-p[j]|/n
Misclassification Rate	0,4692	∑ (p[j]≠pMax)/n
N	844	n

Parameter Estimates

Term	Estimate	Std Error	ChiSquare	Prob>ChiSq
Intercept	-0,0704829	0,1361854	0,27	0,6048
Niewykryte [%]	-0,0014198	0,0031537	0,20	0,6526

For log odds of M/B

Figure 3.18: Matching segmentation results to malignant and benign cases

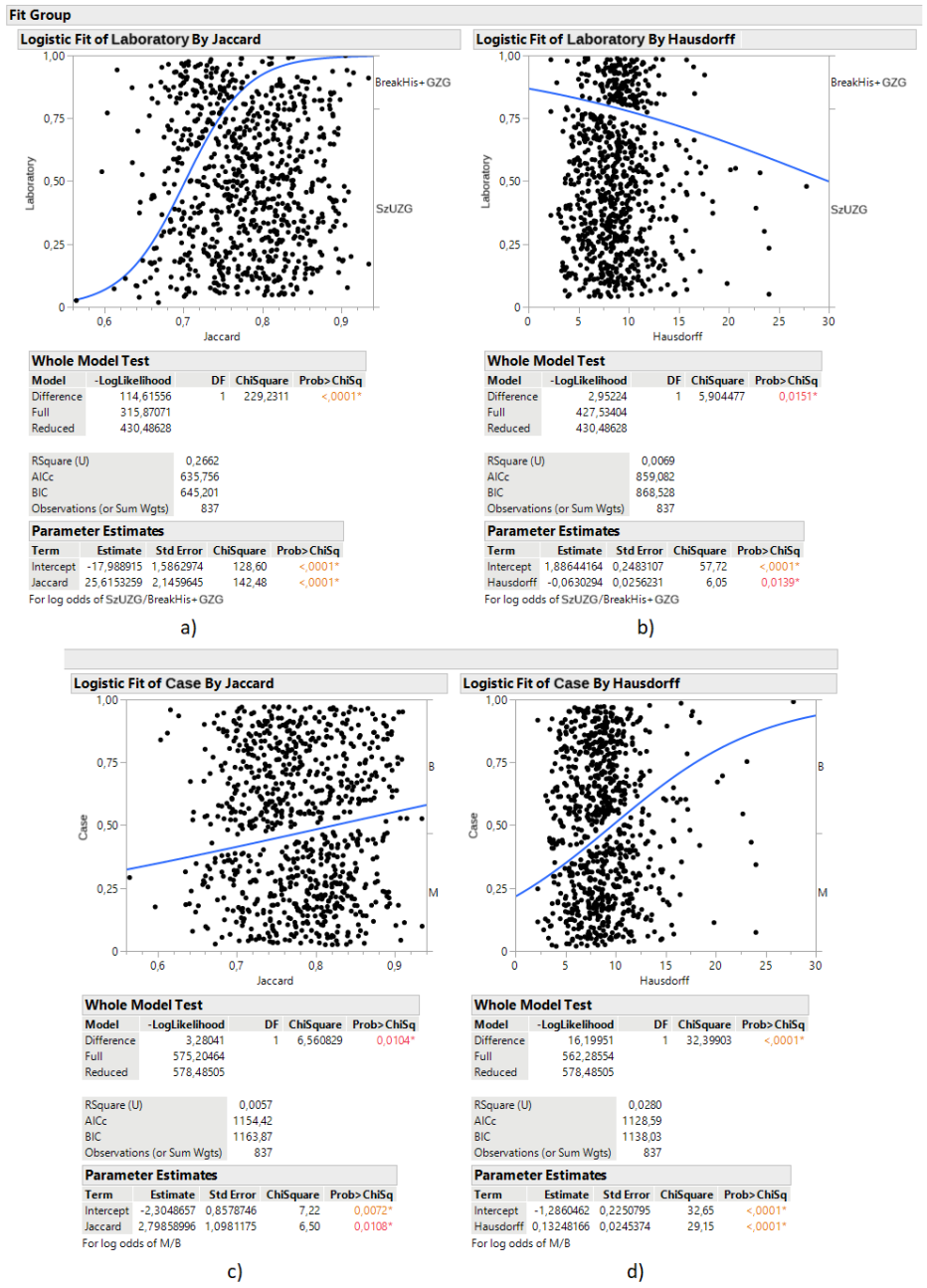


Figure 3.19: Matching the results of segmentation metrics to origin from sets (a, b) and cancer cases (c, d)

Chapter 4

COMPREHENSIVE CLASSIFICATION SYSTEM

4.1 Introduction

Classification is one of the main processes in machine learning and belongs to the group of supervised learning techniques. A classifier is a type of statistical algorithm whose task is to assign observations to appropriate classes based on the characteristics of the examined observation. Therefore, a training set with correctly identified class affiliations of individual observations is necessary in order to carry out a classification process.

4.2 Manual feature extraction

Feature extraction (Fig. 4.1) is a dimensionality reduction method often used to generate features that describe objects in a digital image. The built set of features can then be used to classify images. The extraction of a set of features can be based on domain knowledge. When creating features based on digital images, some programming environments (including the *regionprops* function in Matlab and Python) possess implemented tools that determine the basic properties of objects One of the currently developed grading methods for invasive breast cancer

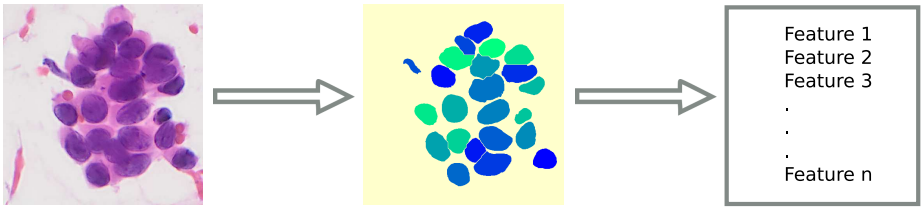


Figure 4.1: Feature extraction

is the Bloom-Richardson scale in the Nottingham modification. The method was published in 1957, and one of the levels of its scale is based on the polymorphic nature of cell nuclei. Their shapes and sizes are assessed:
• cell nuclei are small, uniform in shape and in size,
• cell nuclei are medium or large, but usually of similar shapes and sizes,
• large cell nuclei display a clear discrepancy in shapes and sizes.

In addition, doctors mention such features as:
• distances between neighboring nuclei,
• degree of overlap of cell nuclei.
The line between malignant and benign is very imprecise. Only experience gained in medical practice and unrecognized, difficult to define features of cell nuclei lead to an effective diagnosis.

Segmentation using CNN makes it possible to calculate the basic parameters of detected cell nuclei. In the study, each cell nucleus was characterized by one of 31 parameters. Then, for each image of size 230×230, the statistics of the cell nuclei were counted. These statistics included mean, median, standard deviation, kurtosis, skewness, entropy, k-statistic, and interquartile range. 31 parameters multiplied by 8 statistic categories resulted in 248 object features for each image plus 3 gradient features, which resulted in a total of 251 manual features. Based on these features, an image classification model was built.

The set of characteristics of cell nuclei can be divided into three main subgroups. The first subgroup defines physical parameters of cell nuclei based on a binary object mask. These features include Area, AreaBB (*Bounding Box*), Perimeter, Major Axis, Minor Axis, Eccentric, Diameter corresponding to the circle around the object, Circularity, Aspect (*Object proportions*), Ellipticity, Skeleton.

The second group of features is related to the arrangement of organelles and chromatin inside the cell nucleus. These features can be described as textural and are examined on the basis of the gray level co-occurrence matrix (GLCM (Haralick, Shanmugam & Dinstein 1973)) as well as on the gray level run length matrix (GLRLM (Galloway 1975, Tang 1998)).

The GLCM matrix is created by examining the values of neighboring pixels. Verification involves checking what values neighboring pixels possess. Assuming the simplest case, in which we check two vertically adjacent pixels, we can check how many cases in the entire image space there are when, for example, a pixel with a value of 2 is above a pixel with a value of 152. By counting the number of such cases, we can write it down in row 3 and column 153 (assuming addressing from the value 0). Hence, it is easy to conclude that the space of available comparison pairs is limited by the pixel values, i.e. in the space of an eight-bit, single-channel image, it is the space from 0 to 255. This would result in a 256 by 256 matrix on which the number of occurrences of individual pairs is summed up. Therefore, just before the GLCM matrix is created, the image is converted so that the values in the image range from 0 to 32. This way, it will be easier to find neighboring pixels with similar intensity.

The next stage of creating a GLCM matrix is value normalization, which involves dividing the value of each individual cell of the resulting matrix by the sum of values of all matrix cells. The matrix is then converted to the final version based on averaging the values for the resulting matrices in the directions $0°$, $45°$, $90°$ and $135°$. The last parameter is the distance at which the pairs are examined, which is 5 pixels. Based on the resulting GLCM matrix, its standard statistics, such as contrast, dissimilarity, homogeneity, energy, correlation and second angular moment, were calculated and averaged. The averaged statistics were then calculated into

mean, median, standard deviation, kurtosis, skewness, entropy, k-statistics, and interquartile range characteristics. The final number of features from the GLCM matrix was 48.

The next set of textural features was based on GLRLM (Gray Level Run Length Matrix) (Galloway 1975). GLRLM can be defined as a matrix $N \times M$ p, where N is the number of gray levels and M is the maximum run length, defined for a given image as the number of runs with gray level pixels i and length j. As in the case of GLCM, we calculate the run level matrices for 0°, 45°, 90° and 135° using five levels of gray: SRE (*Short Run Emphasis*) emphasis on the short period of GLRLM, LRE (*Long Run Emphasis*) emphasis on the long period of GLRLM, GLU (*Gray Level Uniformity*) GLRLM gray level heterogeneity, RLU (*Gray Level Uniformity*) GLRLM run length heterogeneity, RPC (*Run Percentage*) percentage of the GLRLM run.

The last group of features is related to the colorimetric features of the nuclei. These are: average value of the red channel (MR), average value of the green channel (MG), average value of the blue channel (MB), average brightness value (ML), variance of the red channel value (VR), variance of the green channel value (VG), variance of the blue channel value (VB), variance of the brightness value (VGR), entropy of the brightness value (ENGR).

A small set of gradient features based on the structural tensor was also prepared (they describe the gradient distribution in the vicinity of a point). These features are difficult to classify into one of the main feature categories. The maximum values of the components of the structural vector are added to the set of features, taking into account the entire image and all calculated tensors. There are 4 components of the vector, 2 of which are of the same value. Ultimately, 3 new features were obtained.

Table 4.1: List of extracted features of cell nuclei

Morphometric features	
Area, AreaBB (Bounding Box), Circumference, Major Axis, Minor Axis, Eccentric, Diameter (equivalent), Circularity, Aspect Ratio, Ellipticity, Skeleton	
Colorimetric features	
Red Channel Average Value (MR), Green Channel Average Value (MG), Blue Channel Average Value (MB), Average Gray Intensity Value (MGR), Red Channel Variance (VR), Green Channel Variance (VG), Blue channel variance (VB), Gray intensity variance (VGR), Gray intensity entropy (ENGR)	
Textural features	
GLCM	GLRLM
Contrast (GLC Contrast), Correlation (GLCM Correlation), Energy (GLCMEnergy), Homogeneity (GLC MHomogenity), Dissimilarity (GLCMDissimilitary), Second angular moment (GLCMASM)	SRE (Short Run Emphasis), LRE (Long Run Emphasis), GLU (Gray Level Uniformity), RLU (Gray Level Uniformity), RPC (Run Percentage)
Gradient features	
Components of the structural tensor: Tensor str arr, Tensor str acr, Tensor str acc	

4.3 Deep feature extraction

A highly effective approach is to extract deep features based on weights obtained for neurons in intermediate layers. During learning, the neural network adjusts the weights of neurons to the response that generates the smallest loss. In the case of binary classification, the last layer may assume the structure of one neuron, which receives information from higher layers. In the proposed approach, it is possible to extract output values of neurons from the trained network when a new image passes through the network. The values obtained in this way constitute sets of deep features.

4.3.1 Machine learning

The main topic discussed in the work is the issue of deep neural networks, which are part of an extensive field known as machine learning, which in turn is part of the field of computer science called artificial intelligence (Fig. 4.2 (Chollet 2017a)).

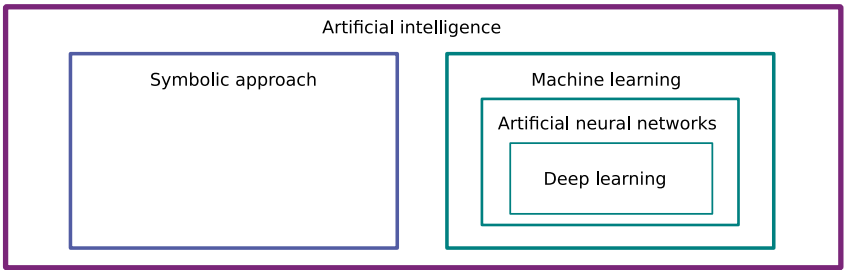


Figure 4.2: Diagram of the field of artificial intelligence according to Chollet

The symbolic approach involves creating logical models of a problem under study in the form of a set of rules formulated and understood by humans. The set of rules, in turn, makes it possible to perform intelligent tasks within the scope of the rules contained in the implementation. The symbolic approach dominated the world from the 1950s to almost the 1990s, but it is still widely used because for many simple tasks the use of machine learning is unjustified. The most interesting examples of the symbolic approach include fuzzy logic and genetic algorithms.

On the other hand, machine learning creates rules based on provided data. It is difficult to fully define in a few sentences what machine learning is, but a few basic issues which appear in the field of machine learning are worth mentioning. One of the basic tasks of machine learning is classification. The task of the classification process is to assign elements from a given set to appropriate classes after taking into account the features of the examined objects. In turn, the features of objects are obtained in the process of feature extraction as well as through reducing the dimensionality to a set of important features. The process of dimensionality extraction and reduction can be fully controlled by a human or, in the case of artificial neural networks, in the form of a "black box" to which previously

prepared training sets are sent, on the basis of which the network interprets the most important features of the examined objects in subsequent teaching eras.

4.3.2 Elements of artificial neural networks

The beginning of the concept of artificial neural networks is widely considered to be rooted in the construction of a perceptron. A decade later, its limitations were defined in that it could only solve linearly separated problems (Minsky & Papert 1969). Criticism of the perceptron resulted in a decline in interest in the technology. Another decade later, the concept of a CNN network and the use of composite cells appeared, an idea similar to modern pooling layers (Fukushima 1980). Training complex neural networks could not be performed without an appropriate training process, so to solve this problem, the backpropagation algorithm was developed (Rumelhart, Hinton & Williams 1986). In addition to the new algorithm, it was necessary to develop a differentiable neuron activation function. A significant number of activation functions that meet the continuity criterion were been developed. Examples of differentiable activation functions are:
• unipolar sigmoid function:

$$y(x) = \frac{1}{1 + e^{-\beta x}} \tag{4.1}$$

• rectified linear unit activation function (ReLU)

$$f(x) = \begin{cases} 0 & \text{for } x \leq 0. \\ x & \text{for x} > 0. \end{cases} \tag{4.2}$$

• hyperbolic tangent:

$$y(x) = tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{4.3}$$

Work on the development of the CNN network (LeCun & Bengio 1995) resulted in combining the ideas of the CNN network with the backpropagation algorithm and obtaining a neural network capable of effective training (Lecun, Bottou, Bengio & Haffner 1998). It can therefore be concluded that the potential of the CNN network in the process of digital image processing was practically used already in the 1990s. However, only a small group of scientists were focused on the development of this technology. At that time, the classical approach dominated, consisting in building a set of features and classifying objects according to a reduced set of features. In the 1990s, it was concluded that CNNs could help eliminate the problem of image preprocessing and object feature extraction (LeCun & Bengio 1995). The current level of interest in deep neural networks follows the success of AlexNet in the 2012 ImageNet competition.

4.3.3 Convolutional neural networks

Introduction

CNN is an example of a deep network organized in a hierarchical manner. Subsequent layers process data from a lower level. Data obtained at a lower level

is transferred to the next layer, thanks to which the network gradually obtains more complex information. CNNs are primarily used in image segmentation and classification. Moreover, they are used in the problem of transfer learning, i.e. a research problem involving the use of networks specialized in solving one problem to solve another problem. An example CNN network is schematically presented in

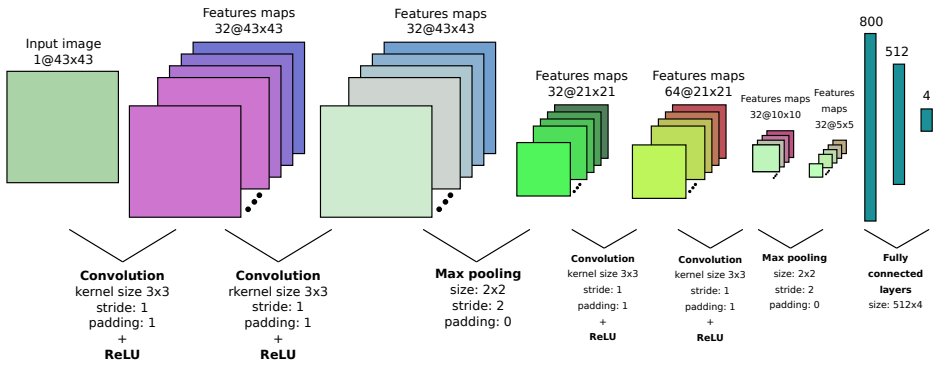


Figure 4.3: Example of CNN

Fig. 4.3. The attached example shows a CNN network intended for image classification. This is a standard task that CNNs are designed for. A digital image of a specific dimension constitutes the input to the network. The network presented in Fig. 4.3 performs a convolution operation in the first processing step. The input image is processed by 32 filters with a kernel size of 3 by 3, a shift value of 1 and a complement value of 1. In the example provided, the ReLU neuron activation function was selected. This operation is followed by another convolution and then a maximizing pooling operation, followed by two more convolutions and then another maximizing pooling operation. Finally, there is a connection with neurons in the hidden layers. The output layer consists of 4 neurons, which means that the network has classified or segmented 4 types of objects in the image.

AlexNet

One of the simplest CNN network architectures is AlexNet, which is also a milestone in the spread of deep learning. The structure of the AlexNet network is shown in Fig. 4.4. It was designed to work with 224×224 images and 3 channels. The depth of the network is 8 layers, including 5 convolutional layers and 3 dense layers (with fully connected neurons). Despite its relatively shallow depth, this network has approximately 62 million parameters. Currently, thanks to the development of graphics card technology, it can be an effective and quick tool for solving classification problems. Moreover, despite the fact that a decade has passed since its debut, the AlexNet is still used (Zhao, Zhao, Yang & Xu 2023).

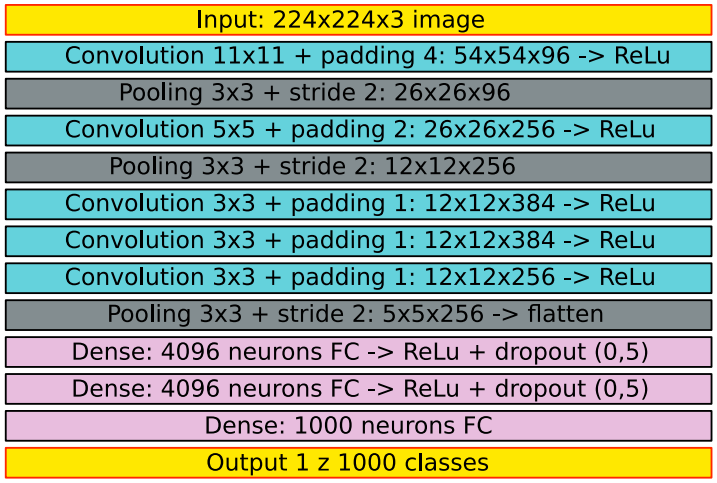


Figure 4.4: AlexNet network

VGG

Depending on the number of their deep layers (16 or 19), the next group of deep neural networks are commonly called VGG16 or VGG19 (Simonyan & Zisserman 2015). While equaling or exceeding other models in the number of parameters, the number of deep layers in these networks is significantly different from most popular deep neural networks. VGG networks are characterized by the use of a fixed 3×3 convolution kernel with shift and padding of 1. In addition, each set of convolutional layers concludes with a maximizing pooling layer with the shift and padding of 2. They also double the depth of their filters after each pooling layer. The VGG16 network (see: Fig. 4.5 and the VGG19 network differ in the

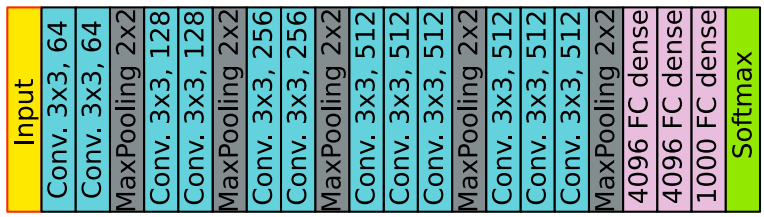


Figure 4.5: VGG16 network

number of the last two blocks of convolutional layers from 3 to 4. The main application of the VGG network is to recognize and classify images. The VGG16 network is 16 layers deep and in its original version it has 138.4 million parameters. The VGG19 network is characterized by 19 layers and 143.7 million parameters.

GoogLeNet

The design assumptions of the GoogLeNet (Szegedy *et al.* 2016) network focus on achieving high performance while maintaining high network depth. As a result, the network is equipped with several innovative solutions. One of these was the introduction of a new subnetwork called the inception module (Fig. 4.6). In

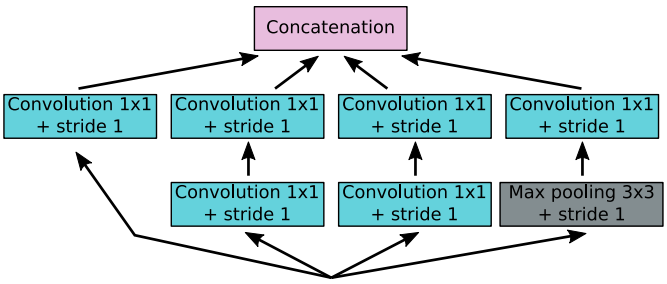


Figure 4.6: Inception module

addition, the GoogLeNet network was equipped with two smaller classification networks, consisting of several layers: an averaging layer, a single convolutional layer, two fully connected layers, and a softmax activation layer. These networks were connected to the third and sixth inception modules, and their task was to increase the generalization capacity of the networks and, above all, to reduce the problem of gradient vanishing (Géron 2020).

ResNet

The ResNet network, or rather the ResNet family of residual networks, was proposed in 2015-2016 in several articles on image recognition (He *et al.* 2016a, He, Zhang, Ren & Sun 2016b). These networks are characterized by a large depth indicated by the number of layers (18-151) and the presence of bypass connections between the layers. The goal of the skip-connection technique is to try to avoid gradient vanishing. ResNet networks have demonstrated high performance and have won several image classification competitions. It can therefore be assumed that these networks will achieve high accuracy in research on the classification of medical images.

DenseNet

DenseNet (Huang *et al.* 2017) was proposed in 2016. This model is characterized by direct connections between all layers to improve information flow (Huang *et al.* 2017). DenseNet networks belong to the group of residual networks, so they largely resemble ResNet networks. The key difference is its denser concentration of layers in the DenseNet network (hence its name) and its method of combining objects (ResNet uses summation, while DenseNet uses concatenation) (Huang

et al. 2017). As with all residual networks, skipping layers is intended to prevent gradient vanishing.

Xception

The Xception network was proposed by Chollet (Chollet 2017b) as an alternative to the large Inception V3 network (an extension of GoogLeNet). The Xception network has the same number of parameters as InceptionV3, but uses model parameters more efficiently (Chollet 2017b). The Xception network can therefore be used as an alternative to the InceptionV3 network. What connects these networks is the use of an inception module. This module is used to reduce the dimensionality of data for more efficient operation and consists of a group of convolutional filters, usually of different sizes. In the Xception network, the inception module has been reduced to an "extreme" form, in which there is one convolutional 1×1 layer, which is then divided into a set of 3×3 elements. Reducing the number of convolutional layers has a positive effect on the performance of the Xception network.

4.3.4 Team learning methods

One way to improve prediction using classifiers is to combine them into ensembles. The idea is that a larger number of weaker models predicting the same problem can increase the accuracy of that prediction. There are several basic ensemble classification methods in the literature, such as: aggregation (bagging), boosting, random forest, as well as voting and contamination (stacking).

Voting

One of the basic examples of building teams of classifiers is voting. This method involves making a classification decision based on majority voting. Voting result $\hat{y}$ obtained on the basis of training data (x) used when building the classifier model (C):

$$\hat{y} = mode\{C_1(x), C_2(x), ...C_n(x)\} \tag{4.4}$$

Most often, ensembles consist of heterogeneous classifier models using a set of training data. The purpose of the variety of models used is to eliminate systematic errors. A modification of majority voting is a weighted voting method, in which the classifiers have different weights, while the obtained result is based on a weighted average.

Aggregation (bagging)

Aggregation method is another approach to solving the problem of an ensemble of classifiers. It involves drawing and returning many samples of training data. The randomly selected samples take part in tuning similar classifier models. As in the case of the voting method, the final classification result is obtained as a dominant result among the component models.

Random forests

Random forests are an extension of the aggregation method. In this method, decision trees are in the function of base classifiers. Each tree is modeled on a random training set. The final decision is made by majority voting.

Boosting

The idea of boosting is to build a sequential aggregate strong classifier on the basis of weak classifiers in such a way that the next model tries to improve the result of the previous classifier (Géron 2020). Boosting can take the form of one of the basic algorithms, e.g.: adaptive boosting (AdaBoost) or gradient boosting.

Contamination (stacking)

The contamination concept differs from other ensemble methods in that instead of a simple voting function, another superior classifier is used to make decisions.

Deep feature generator

The work proposes a procedure for building deep feature generators inspired by the issue of classifier ensembles. The general idea is to strengthen classifier models by combining them. The work proposes a procedure for building deep feature gener-

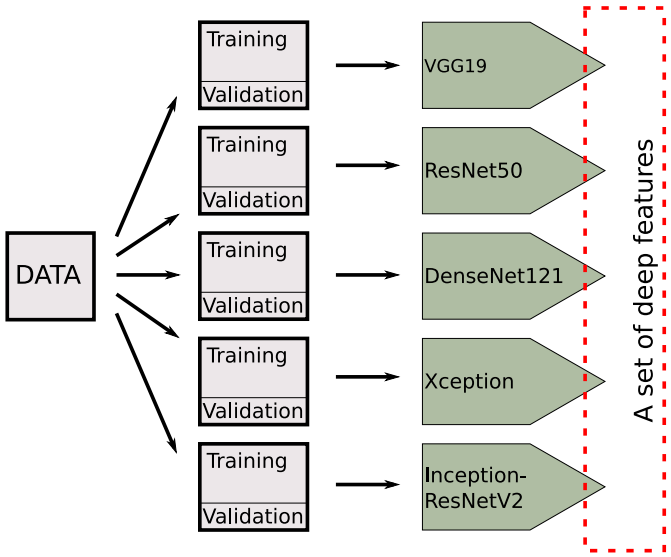


Figure 4.7: Deep feature generators

ators inspired by the issue of classifier ensembles. The general idea is to strengthen classifier models by combining them. The proposed method is mainly based on ensemble methods, where there is a random set of data and a heterogeneous set of

classifiers. In this case, the set of classifiers is replaced by deep feature generators. The classification decision is not made by the results of individual classifiers, but on the basis of deep features obtained from the penultimate layer of neurons. In order to diversify the team, neural networks, such as: ResNet, DenseNet, VGG, InceptionResNetV2 and Xception, were selected to represent various groups of networks. The heterogeneity of the network is intended to ensure a variety of features, similar to the method of an ensemble of voting classifiers.

Each of the neural network models included in the classifier team in the penultimate layer received 5 neurons with the ReLU activation function. This limitation did not negatively affect the quality of the obtained accuracy results. Moreover, using only 5 neurons in the penultimate layer resulted in obtaining 25 deep features out of the 5 tested networks. The total number of deep features therefore does not drastically change the total space after feature fusion (251 manual and 25 deep features). The generated set of deep features is characterized by high correlation within individual classifiers and much lower correlation between classifiers. Unfortunately, the nature of the obtained distributions of deep feature values has a large impact on the correlation, as neurons tend to obtain zero values. The distributions of deep features are therefore characterized by an inflation of zero values. The use of other activation functions does not solve the problem, and the obtained classification results are slightly worse.

4.4 Dimensionality reduction

Dimensionality reduction is an important task in the classification process. The need to reduce features stems from multiplicity, which in turn is rooted in the phenomenon labelled as the curse of dimensionality. It is related to increased complexity of a classifier caused by the increase in the number of parameters, the number of which, in turn, results from the size of the feature vector of the examined objects. Too many features may also cause the effect of overfitting of the classifier to the training set. The phenomenon of overfitting reduces the obtained results on the test set. Dimensionality reduction methods can be divided into two main groups: feature selection methods and feature projection methods.

4.4.1 Extraction and construction of new features

Feature extraction is also classified as a dimensionality reduction method (e.g. calculating the area or perimeter of a cell nucleus is a reduction from image space to a specific feature).

$$\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ \vdots \\ c_n \end{bmatrix} \rightarrow \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} = f\left(\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ \vdots \\ c_n \end{bmatrix} \right)$$

Image conversion to shades of gray or image binarization are also dimensionality reduction methods. They are such common elements of image pre-processing that sometimes they go unnoticed. Also, feature extraction often functions as an introduction to further reduction, for example by means of feature selection methods. In addition to extraction, there are also other feature projection methods included in the dimensionality reduction methods, such as Principal Component Analysis (PCA). This method involves creating a w × r dimensional transformation matrix that will enable the projection of a w-dimensional space into a smaller r-dimensional one (Raschka & Mirjalili 2019). The task of PCA analysis is to transform the generated matrix in such a way as to maximize the variance of the first and subsequent components. PCA can be performed on normalized input variables. Then it is possible to use PCA based on the covariance matrix. The problem of different-scale input data can be circumvented by using a PCA model based on a correlation matrix. Then the model is not sensitive to the different scale of the data. Domain knowledge, which is based on the experience of diagnosticians as well as on the implementation of solutions that diagnosticians notice when examining a sample, can also be used to construct new features.

4.4.2 Feature selection

Feature selection is one of dimensionality reduction methods. The scope of its application is related to building regression models that are sensitive to an excessive number of input features. This fact does not contradict the possibility of using feature selection in other classification models. Feature selection can be divided into three main groups of methods: wrapper, filter and embedded.

Wrapped methods

Wrapped methods (Fig. 4.8) require further training steps to obtain the appropriate subset. The problem is basically limited to the problem of finding a subset of features. The disadvantage of these methods is their computational cost. For-

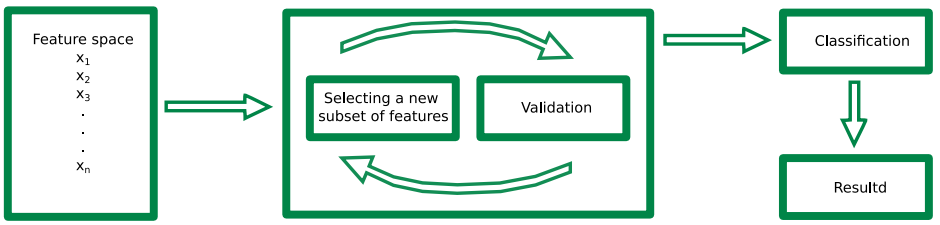


Figure 4.8: Feature selection - wrapped methods (wrappers)

ward selection is a selection method that involves adding further features. The algorithm starts with an empty set of features, then uses a selected criterion (e.g. Akaike (Akaike 1998), Bayesian Shwarz (Schwarz 1978)) to check whether the added feature improves the value of the criterion. It is also possible to examine the importance of features in the model and add those features that meet the required

confidence level. Adding further features is done iteratively, so for example, a selected number of subsequent iterations that do not improve the validation result may constitute a stopping point for the algorithm.

Backward selection is a feature selection method analogous to forward selection, with the difference that the algorithm starts with a model containing all the features. Iteration, on the other hand, involves discarding the least important feature until the model stops improving.

Mixed selection is a selection method combining forward and backward approaches. The algorithm may reject or add a feature depending on whether such action will improve the quality of the model.

Filtering methods

Filtering methods (Fig 4.9) are usually used as a pre-processing step. Feature selection is independent of machine learning algorithms. Instead, features are selected based on their performance in various statistical tests. Typically, Pearson's

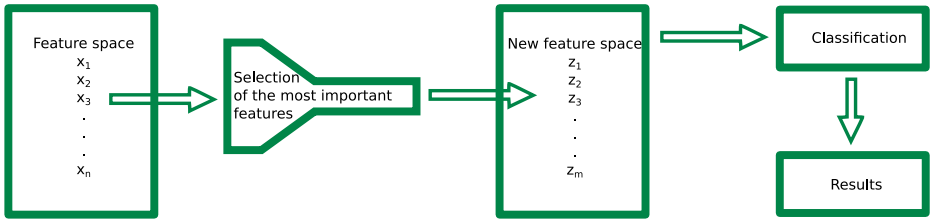


Figure 4.9: Feature selection - filtering methods

linear correlation (Pearson 1895) is a sufficient filter to eliminate features in small feature sets, which is expressed by the formula:

$$r_{xy} = \frac{\Sigma(x_i - \bar{x})(y_i - \bar{y})}{(\Sigma(x_i - \bar{x})^2)(\Sigma(y_i - \bar{y})^2)}, \tag{4.5}$$

where r_{xy} is the correlation coefficient of variables, x_i is a single observation from set x, $\bar{x}$ is the average value of observations in set x, x_i is a single observation from set x, $\bar{x}$ is the average value of observations in the set x. Therefore, if the variable x is highly correlated with the variable y, one of these features can be eliminated from the set. This analysis is effective for smaller feature sets. However, it is worth examining larger sets of features in terms of their association with a larger number of features at the same time. The variance inflation factor (VIF), defined by the formula:

$$VIF_m = \frac{1}{1 - R_m^2}, \tag{4.6}$$

where R^2 means the coefficient of determination for factor m is an ideal tool for this purpose. The coefficient of determination is used to determine the fit of the

model to the data and is set in the range from 0 to 1 (the higher the value, the better the explanation of the variable). In other words, R^2 indicates how much of the response variable is explained by the model. Therefore, if the R^2 coefficient is calculated for a specific variable, it can be seen that the more this variable is explained by other variables, the higher the R2 value and, consequently, the VIF. It is assumed that variables for which the VIF coefficient exceeds 10 should be removed from the model. A variable that obtains a value of 10 for the VIF coefficient is explained by 0.90 by the remaining variables. These coefficients can be used for continuous variables. However, if the data for the model are nominal, then the χ^2 independence test may be an appropriate filter:

$$\chi^2 = \sum_{i=1}^{m} \sum_{i=j}^{n} \frac{(n_{ij} - \hat{n_{ij}})^2}{\hat{n_{ij}}}, \tag{4.7}$$

where $n_i j$ means empirical numbers and $\hat{n_i} j$ means theoretical numbers.

Embedded methods

Embedded methods (Fig. 4.10) are a group of techniques in which the selection of features is the result of the model building process. The performance of built-in

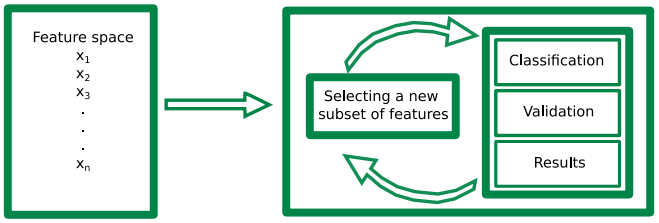


Figure 4.10: Feature selection - embedded methods

methods can be explained using linear regression. The linear regression model is trained until the sum of squared residuals between the true and predicted values (called RSS or SSE in the literature) is minimized (Albon 2019). The sum of squares in question is expressed by the formula:

$$SSE = \sum_{i=1}^{m} (y_i - \hat{y_i})^2, \tag{4.8}$$

where y_i denotes actual values and $\hat{y_i}$ denotes predicted values. Regression models can be regularized. In such a case, a regularization term is added to the cost function(Géron 2020). The purpose of regularization is to reduce the variance of the model (Albon 2019).

Lasso (Tibshirani 1996) is a popular method for regularizing regression models. The name of the regularizer comes from the English words *least absolute selection and shrinkage operator*.

$$SSE_L = \sum_{i=1}^{m}(y_i - \hat{y_i})^2 + \lambda \sum_{j=1}^{n} |\beta_j| \tag{4.9}$$

The Lasso method is therefore similar to the ridge method with the difference that when multiplying by the degree of regularization, instead of the sum of the squares of the weights, it takes into account the sum of the absolute values of the model weights.

A slightly newer approach called Elastic Net (Zou & Hastie 2005) combines both of the above-mentioned regularization methods, giving the formula:

$$SSE_E = \sum_{i=1}^{m}(y_i - \hat{y_i})^2 + \lambda \sum_{j=1}^{p} \beta_j^2 + \lambda \sum_{j=1}^{n} |\beta_j| \tag{4.10}$$

When there are strong correlations between several features, according to (Géron 2020), the flexible grid method is more effective than the Lasso method.

In addition to methods with built-in feature selection, the L2 regularization method should be mentioned. It is characterized by the lack of built-in feature selection. L2 regularization is also known as ridge regression (Hoerl 1962):

$$SSE_G = \sum_{i=1}^{m}(y_i - \hat{y_i})^2 + \lambda \sum_{j=1}^{n} \beta_j^2, \tag{4.11}$$

where the hyperparameter λ specifies the degree of regularization (Géron 2020). A λ value of 0 means no regularization. If the value of the hyperparameter λ is high, it means that the model weights (β_j) will be close to 0 (Géron 2020).

4.4.3 Stochastic feature selection

In the conducted research, the highest average accuracy results for the classification of cytological and histopathological images were obtained for logistic regression models using the forward feature selection method and the elastic net regularization method. In the case of forward feature selection, practical problems which occurred when testing the data on the built models were related to the relatively poor classification of images containing malignant cases, with the simultaneous above-average correct classification of benign cases. Models created using elastic net regularization are, on the other hand, difficult to interpret due to insufficient reduction of model dimensionality. Both methods mentioned above achieved similar classification accuracy results of just over 0.78. An alternative solution to the feature selection problem may be based on the use of stochastic methods. However, a thorough search involves high computational costs. This fact contributed to the proposal of an approach that would limit the number of potential features needed to obtain an accurate classification model.

The assumption is to randomly select the length of the feature vector used in training the logistic regression model. However, the vector can contain any set of features, hence the power of the elementary event space $|\Omega|$ is expressed by a formula:

$$|\Omega| = \sum_{k=1}^{n} \frac{n!}{k!(n-k)!}, \tag{4.12}$$

where n denotes the total number of features and k denotes the length of the feature vector. The great number of possibilities means a huge amount of time needed to search through such space, so some restrictions will be applied in the next step.

It can be assumed that the length of the feature vector affects the probability of obtaining a high classification result. To verify the assumption, an experiment was performed to draw 100 sets of features with a random length from the range of 1 to $n-1$, because the set of all n features had been tested in previous experiments. From all 100 sets, the one with the highest classification result on randomly extracted validation data was selected. The entire experiment was repeated 100 times in order to obtain an empirical distribution of the lengths of the feature vectors of the best sets. The obtained empirical distributions are characterized

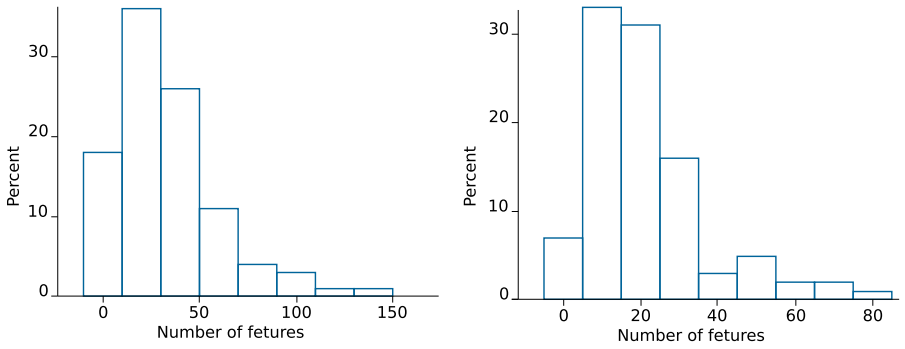


Figure 4.11: Examples of empirical distributions

by right-sided asymmetry, resembling the gamma distribution. The use of the theoretical gamma distribution requires the calculation of the parameters of this distribution. The method of moments can be used to estimate parameter values. Assuming x_i is a single observation, where $i = 1, 2, ..., n$; furthermore, if the set $X \sim Gamma(\alpha, \theta)$, then the value $E(X) = \frac{\alpha}{\theta}$ and $E(X^2) = \frac{\alpha(1+\alpha)}{\theta^2}$. The method will therefore consist in solving the system of equations:

$$\begin{cases} \frac{\hat{\alpha}}{\hat{\theta}} = \mu_I \\ \frac{\hat{\alpha}(1+\hat{\alpha})}{\hat{\theta}^2} = \mu_{II} \end{cases} \tag{4.13}$$

When substituting μ_I into the second equation we obtain:

$$\left(\frac{1}{\hat{\alpha}} + 1\right) \mu_I^2 = \mu_{II}, \tag{4.14}$$

then:

$$\frac{1}{\hat{\alpha}} = \frac{\mu_{II}}{\mu_I^2} - 1 \Rightarrow \hat{\alpha} = \frac{\mu_I^2}{\mu_{II} - \mu_I^2} = \frac{\bar{X}^2}{\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^2} \tag{4.15}$$

and substituting $\hat{\alpha}$ into the second equation:

$$\hat{\theta} = \frac{\hat{\alpha}}{\hat{\mu}_I} = \frac{\bar{X}}{\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^2}, \tag{4.16}$$

where $\bar{X}$ means the average, and $\frac{\bar{X}}{\frac{1}{n}\sum_{i=1}^{n}(X_i-\bar{X})^2}$ variance. Very often the θ parameter is transformed into the scale parameter β:

$$\beta = \frac{1}{\theta} \tag{4.17}$$

hence, we can finally obtain the shape parameter α and the scale parameter β:

$$\alpha = \frac{\bar{X}^2}{V[X]} \qquad \beta = \frac{V[X]}{\bar{X}}, \tag{4.18}$$

where $V[X]$ is the sample variance. In practice, the use of gamma distribution makes it possible to limit the set of searched lengths of the feature vector. The entire feature selection process consists of two main parts, with the first part described by means of a pseudocode marked as Algorithm 1. In the presented algorithm, the dimension of the feature vector marked as $dim(x)$ is of key importance.

Algorithm 1 Determining the parameters of the gamma distribution

Initialize: $x\_Best = 0$; $ACC = 0$; $x\_List = []$
Input: File with classification data
Output: α, β
for $i = 1$ to 100 **do**
    Pick a number x from 1 to $n - 1$
    for $j = 1$ to 100 **do**
        Draw a set of x random features from the range 1 and $n - 1$
        Select the validation and training data set in a ratio of 0.2:0.8
        Build a logistic regression model on the training set
        Test the model on the validation set and calculate ACC
        if $ACC >$ ACC_Best **then**
            $ACC\_Best \leftarrow ACC$
            $dim(x\_Best) \leftarrow dim(x)$
        end if
    end for
    x_List.append(dim(x_Best))
end for
$\alpha \leftarrow mean(x\_List)^2/variance(x\_List)$
$\beta \leftarrow mean(x\_List)/variance(x\_List)$

After calculating gamma distribution parameters, we move on to the part responsible for stochastic feature selection (Algorithm 2). To increase the generalization potential of the randomized models, 5-fold cross-validation and L2 regularization were used, which, unlike L1, does not have a built-in dimensionality reduction. The number of iterations of the algorithm can be set manually. It would also be worth considering to terminate the algorithm after a set number of iterations with no improvement in the accuracy score (ACC).

Algorithm 2 Stochastic feature selection

Initialize: $x\_Best = 0$; $ACC = 0$; $ACC\_Best = 0$
Input: File with classification data, α, β
Output: ACC, x_Best
for $i = 1$ to k **do**
    Randomly select a number x from 1 to $n-1$ based on the Gamma distribution with parameters α and β
    if $x >$ n-1 **then**
        $n - 1 \leftarrow x$
    end if
    Randomly select a set of x numbers denoting features in the range 1 and $n-1$
    Build a logistic regression model with 5-fold cross-validation and L2 regularization
    Calculate ACC for validation data
    if $ACC >$ ACC_Best **then**
        $ACC\_Best \leftarrow ACC$
        $x\_Best \leftarrow x$
    end if
end for

4.5 Master classifiers

This section will briefly present parent classifiers used in empirical studies on classification accuracy using deep manual feature sets and feature fusion. Master classifiers serve as an output subsystem within the developed breast cancer image classification system (Fig. 1.1). Depending on the system version, its input includes manual features, deep features or a set of combined manual and deep features. The output of the parent classifier provides an answer as to whether the cancer case is benign or malignant. In the study, comprehensive experiments were performed to verify the effectiveness of the following classifiers: DT, RL, k-NN, SVM, SN, RF, boosted tree (BT), NB and the classification potential of methods based on discriminant analysis not included in the results. Most of the classifiers used were subject to 5-fold cross-validation. In the case of regression methods, better results were obtained when validating by means of information criteria.

4.6 Classification with an artificial convolutional neural network

In the wake of popularization of ready-made packages for building artificial neural networks, it can be concluded that deep the CNN network is the simplest and one of the most effective classification methods. The most popular neural network models for image classification were prepared in the Keras package, including: ResNet50 (He *et al.* 2016a), VGG16 (Simonyan & Zisserman 2015), Xception (Chollet 2017b). Basically, preparing a model for training involves defining several basic properties, such as: the size of the input image, presence or absence of upper layers of the network, using or not pre-trained weights, using a combining layer, etc. Computational requirements and time-consumption, both in preparing the appropriate database and the calculations themselves, are the two disadvantages of the approach.

4.7 Evaluation methods

There is one standard, basic quality assessment tool among binary classification evaluation methods, i.e error matrix (Tab. 4.2). The table contains two columns and two rows for the predicted class and the actual class. It shows only a fraction

Table 4.2: Confusion matrix

		Predicated class		
		Positive	Negative	
Real class	Positive	TP - True positive	FN - False negative	TPR - Sensitivity $\frac{TP}{TP+FN}$
	Negative	FP - False positive	TN - True negative	TNR - Specificity $\frac{TN}{FP+TN}$
		F-score $\frac{2TP}{2TP+FP+FN}$	ACC - Accuracy $\frac{2TP+TN}{TP+TN+FP+FN}$	PPV - Precision $\frac{TP}{TP+FP}$

of the number of coefficients that can be obtained from it, which are, however the most important from the point of view of the experiments in question. The Accuracy Coefficient (ACC), which determines how well a classifier fits the data, will be of key importance for the experiments. From this coefficient, a classification error can also be determined as a complement of the accuracy to the value of 1. F-score is an alternative coefficient indicating the overall quality of the segmentation model. Other important indicators are sensitivity (TPR) and specificity (TNR). The TPR rate will indicate how well the positive cases have been classified, while the TNR will generate information about the classification performance of true negative cases.

4.8 Results

4.8.1 Scheme of the empirical research conducted

The results presented in this section concern two main approaches (Fig. 4.12). The first part of the results (Chapter 4.8.2) examines classification accuracy of single images using deep CNNs. The experiment began with an examination of images in the RGB space along with a description of the applied model modifications. The rest of the experiment also contained classification results of selected CNN networks on images normalized by means of the H&E deconvolution method. Then, a new normalization method was used based on the developed hybrid cell nuclei segmentation algorithm.

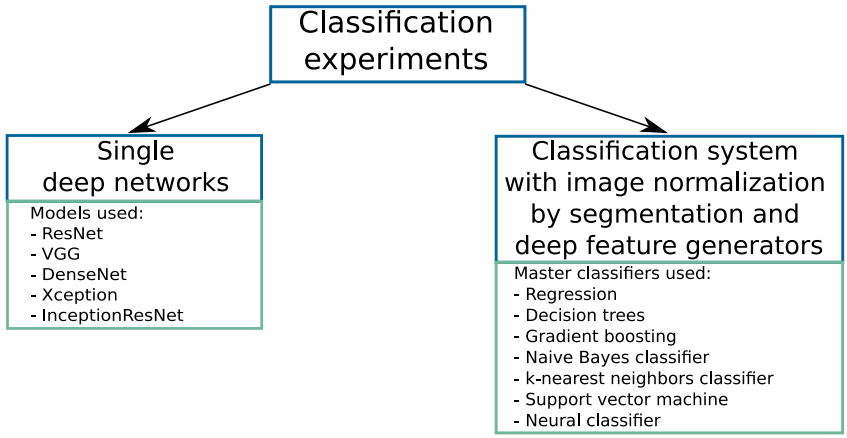


Figure 4.12: Schemat głównych eksperymentów

The second part of the research (Chapter 4.8.3) consisted in verifying classification accuracy on a set of manual and deep features and feature fusion using a comprehensive classification system developed as part of the work (Fig. 4.12). The study was performed for various parent classifiers, which are defined in Chapter 4.5.

Classifying individual small images provides valuable information about classification accuracy. However, the full picture of classification results will be completed by examining the best classifiers based on a final test verifying classification at the patient level

4.8.2 Deep networks

4.8.2.1 Experiment plan

Experiments on CNN networks consist of three main parts. The first part involves training the network based on data in the RGB space. This part also describes the structure of the models, the modifications made and the learning process. Experiments were performed for 7 different deep networks (ResNet50, ResNet152, VGG16, VGG19, DenseNet121, Xception and InceptionResNetV2).

The second part contains results of training and testing the network on images after normalization using H&E deconvolution. The number of networks in the experiment was reduced to five (ResNet50, VGG19, DenseNet121, Xception and InceptionResNetV2). The results were presented in the form of a concise description and a table.

The third part presents the results obtained in experiments carried out on the basis of images after normalization using the proposed segmentation method. In this part, experiments were performed for five deep network models. They are summarized in a table.

The diagram of the planned experiments using the CNN network is shown in Fig. 4.13.

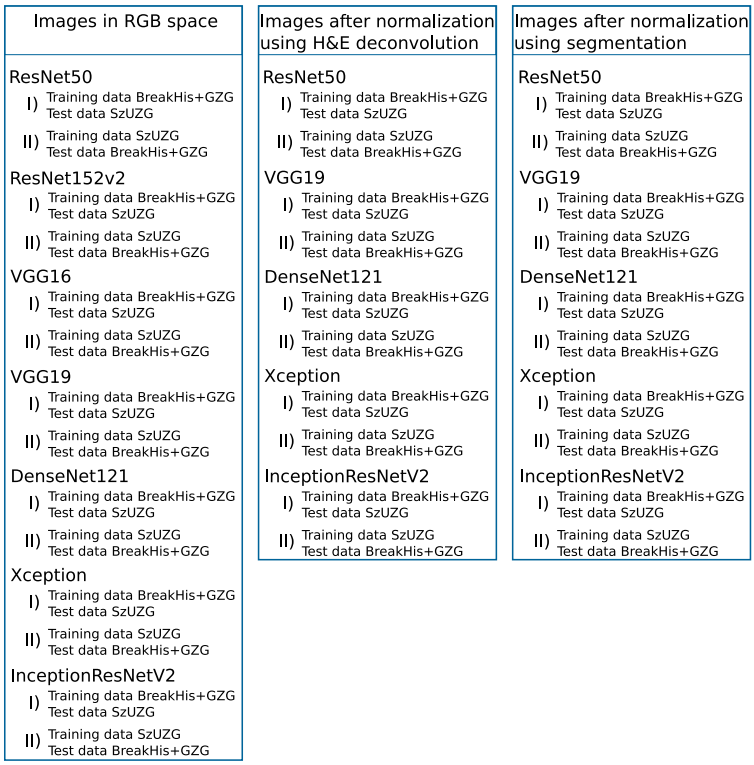


Figure 4.13: Scheme of experiments on CNN networks

4.8.2.2 Introduction

The applied models did not use pre-learned weight values and their upper layers were removed in order to build their own configuration. Additionally, the first layer, which modifies the range of pixel values in the image channels from (0... 255) to (-1... 1), was added to the model. The datasets consist of 230×230 images. Two

sets were created: 1. BreakHis + GZG 12255 images; 2. SzUZG 10071 images. Due to the fact that the number of training data in both sets was relatively large, both sets were used to build separate models, for which data from the second set of images were used to test them. This approach will show to what extent a model built on data from one medical center can be used in the classification of data from other centers. In other words, it will reveal whether the obtained solution is universal and has high generalization properties.

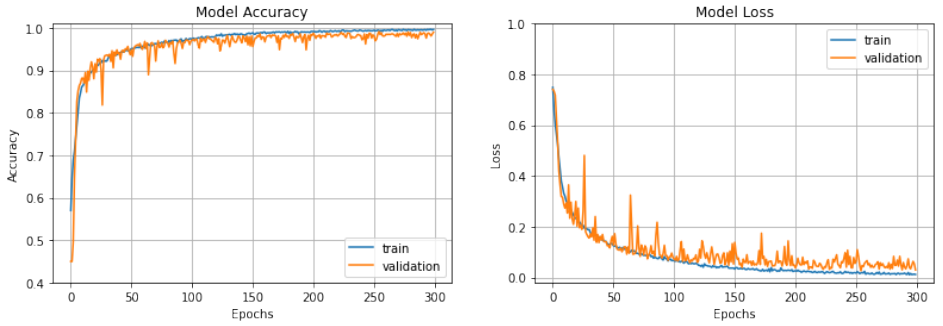
4.8.2.3 Experiments

Images in RGB space

ResNet The study was initiayed with a 50-layer-deep ResNet50 network. All layers were retrained on the training data without using initial weights from other data sets. The input image was resized to match the training images ($230 \times 230 \times 3$). The top layer was removed because it consisted of one Dense layer containing 1000 neurons and replaced with a set of Flatten, Dense(512), Dropout(0.5), BatchNormalization and Dense(1) layers, because the classifier selects one of two image classes (benign, malignant). Additionally, a layer that rescales the input images from pixel values (between 0 and 255) to range (from -1 to 1) was added at the very beginning. Augmentation in the form of rotations and mirror images was also performed on an ongoing basis. No rescaling of size was used because the size of cell nuclei is an important diagnostic factor.

The first training set contained data from the BreakHis + GZG set. 20% of the data (2451) were extracted from the set and transferred to the validation set. The ResNet50 model was used for training, with the top set of layers including: the Flatten layer, the Dense layer with 512 neurons with the ReLU activation function, the Dropout layer (0.5), the Batch-Normalization layer and the final Dense layer with one neuron with a sigmoid activation function. The best model was selected in 255 training epochs based on obtaining the lowest loss on the validation data of 0.0264. The remaining parameters obtained in epoch 255 are: accuracy 0.9944, loss 0.0175 for the training data, and accuracy 0.9914 for the validation data. The result obtained on the test data was 0.5242, i.e. almost complete lack of classification of the test data, with an error of 3.2591. The course of model training (Fig. 4.14) is characterized by stability throughout the subsequent training epochs. Overall, the validation data in the model training process exhibit greater fluctuations in values across successive epochs compared to the training data. However, validation data indicate that the fluctuations decrease in subsequent epochs.

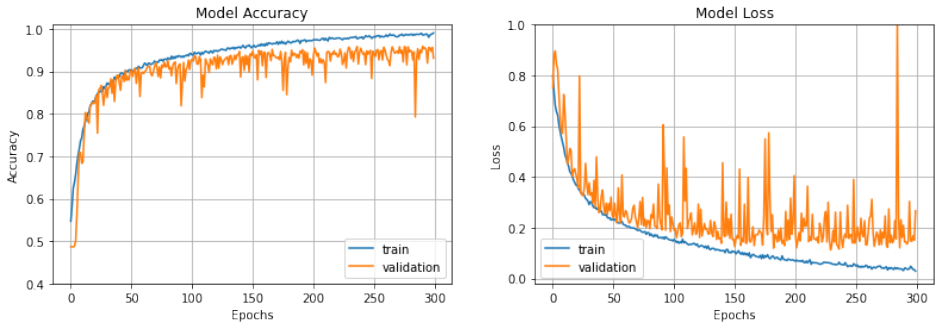
The next experiment involved swapping the test set with the training set. As a result of this operation, a training and validation set was created based solely on data from the SzUZG set, while the test set was created based on the BreakHis + GZG set. The validation set contained 2014 (20%) images extracted from the training set. The best model was selected in the 230 training epoch based on the lowest obtained loss on the validation data of 0.1123. The remaining parameters



(a) Accuracy of the model in subsequent epochs **(b)** Loss of the model in subsequent epochs

Figure 4.14: Training process of the ResNet50 model for the BreakHis + GZG training set

obtained in epoch 230 were: accuracy 0.9767, loss 0.0612 for the training data, and accuracy 0.9568 for the validation data. The result obtained on the test data was 0.6003, while the loss was 1.8699. The course of model training (Fig. 4.15) shows much greater fluctuations in subsequent epochs for the validation set, while the learning process on training data is very stable. The effect of overfitting is also more pronounced here, i.e. after exceeding the 230th training epoch, when the loss for the validation data began to increase with further error reduction for the training data.

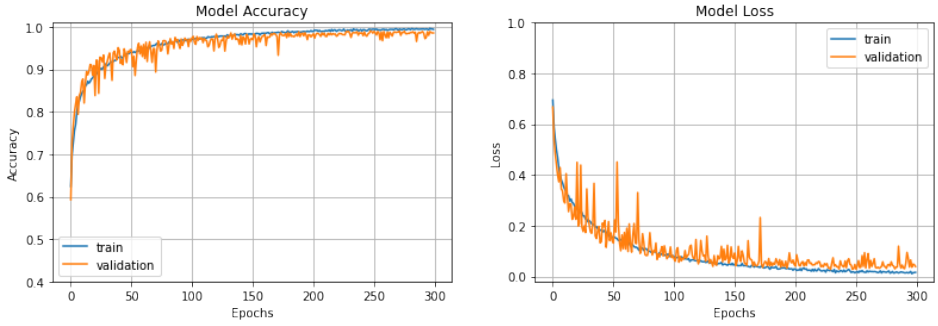


(a) Accuracy of the model in subsequent epochs **(b)** Loss of the model in subsequent epochs

Figure 4.15: Training process of the Resnet50 model for training data from the SzUZG set

The largest network in the group was marked as ResNet152V2. The name indicates that the network has 152 deep layers. As with the smaller ResNet, the top set of layers includes: a Flatten layer, a Dense layer with 512 neurons with a ReLU activation function, a Dropout (0.5) layer, a Batch-Normalization layer, and a final Dense layer with one neuron with a sigmoid function activation. For data

from the BreakHis + GZG set, the lowest loss result on the validation data was 0.0275 in the 230th training epoch, with the accuracy of 0.9898 for the validation set. For the training set in the 230th epoch, a loss of 0.0241 was obtained with accuracy of 0.9916. Unfortunately, the loss for the test data was 3.6754 and the accuracy was 0.5151. The course of the learning curves (Fig. 4.16) is very similar to that of the ResNet50 network. There are only slightly larger fluctuations in values for individual teaching epochs. In the initial training phase, the loss and accuracy results on the validation data do not differ significantly from the results on the training data, but from approximately the 200th training epoch, a separation of the curves begins to appear.

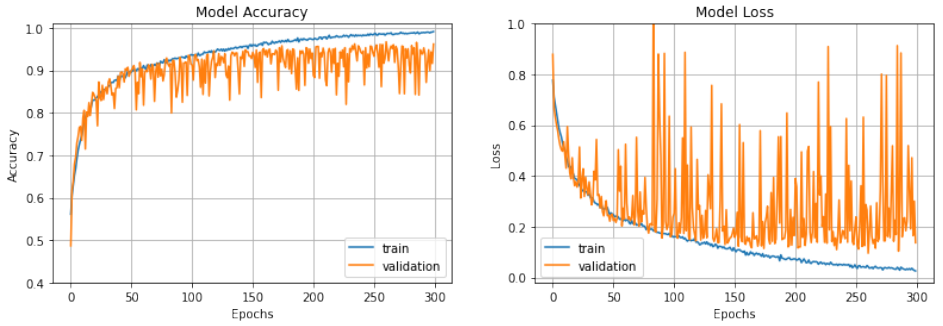


(a) Accuracy of the model in subsequent epochs **(b)** Loss of the model in subsequent epochs

Figure 4.16: Training process of the Resnet152V2 model for the BreakHis + GZG training set

When learning on the SzUZG set, The ResNet152V2 network obtained the lowest loss result for the validation data in the 286th training epoch, amounting to 0.1034, with accuracy estimated at 0.9657. For the training data, the loss value in the 286th epoch was 0.0351 and the accuracy was 0.9881. The loss for the test data was 1.6730 with the accuracy of 0.6562. The obtained learning curve (Fig. 4.17) for the SzUZG set on the validation data is characterized by significant fluctuations, while the learning curve for the training data is characterized by an exemplary course. Around the 150th training epoch, the curves start to move away from each other, which may indicate progressive overfitting to the training data.

VGG16 and VGG19 The preparation of the VGG16 and VGG19 models involved removing the upper layers and replacing them with our own system. No pre-learned weights were used, so the models were pre-initialized randomly. Additionally, the main part of the model was preceded by layers with data augmentation. The upper layers were built from scratch and contained Flatten, two Dense layers with ReLU activation with 4096 neurons (2048 in the case of VGG16 or 1024 in the case of VGG19), then Dropout with a value of 0.5, then there was the Normalization layer, and finally the Dense layer with one neuron with a sigmoid activation func-

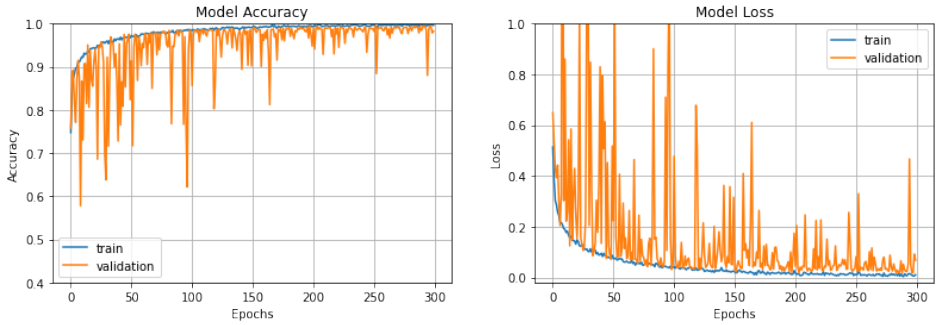


(a) Accuracy of the model in subsequent epochs **(b)** Loss of the model in subsequent epochs

Figure 4.17: Training process of the Resnet152V2 model for training data from the SzUZG set

tion. The total number of parameters for the VGG16 network was 125 876 033 and 126 987 329 for VGG19. The VGG19 network received half as many neurons in the penultimate Dense layer due to exceeding the graphics card memory limit.

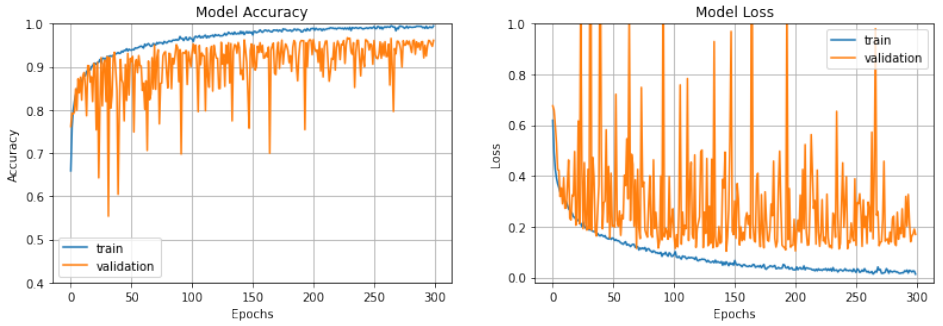
Based on the lowest loss on the validation data of 0.0168, the best model was selected in the training epoch 289. The remaining parameters obtained in epoch 289 were: accuracy 0.9982, loss 0.0063 for the training data, and accuracy 0.9943 for the validation data. The result obtained on the test data was 0.5020 and loss 6.0449.



(a) Accuracy of the model in subsequent epochs **(b)** Loss of the model in subsequent epochs

Figure 4.18: Training process of the VGG16 model for training data from the BreakHis + GZG set

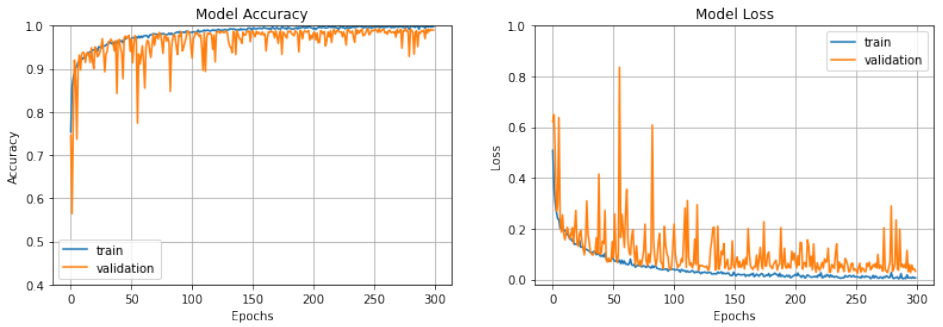
In the case of training the VGG16 model on training data from the SzUZG set the lowest loss for the validation data was 0.1035 in the 144th training epoch. The accuracy of the model for the validation data reached the level of 0.9623, while for the training data the loss was 0.0618 with an accuracy of 0.9746. The test data shows a loss of 1.6542 and an accuracy of 0.7359.



(a) Accuracy of the model in subsequent epochs **(b)** Loss of the model in subsequent epochs

Figure 4.19: Training process of the VGG16 model for training data from the SzUZG set

The VGG19 model achieved the lowest error value on the validation data of the BreakHis + GZG set in the 252th training epoch and was 0.0241 with an accuracy estimated at 0.9918. The training data, in turn, was characterized by a loss of 0.0112 with an accuracy of 0.9968. Unfortunately, the test data again revealed low accuracy results of 0.5032 while the loss was estimated at 6.4768.

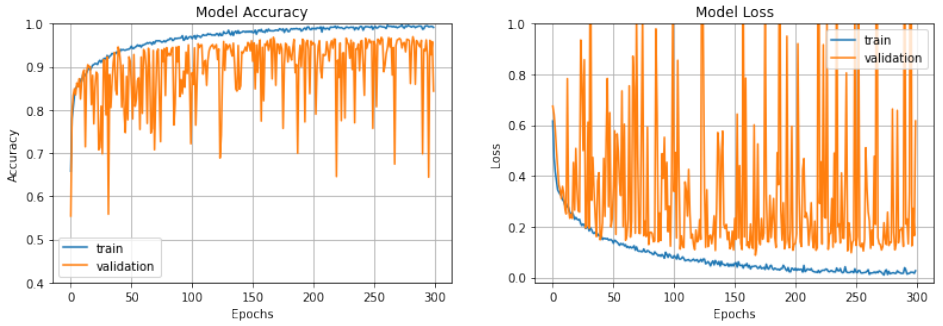


(a) Accuracy of the model in subsequent epochs **(b)** Loss of the model in subsequent epochs

Figure 4.20: Training process of the VGG19 model for the BreakHis + GZG training set

For the training data from the SzUZG set, the best model was recorded in the 168th training epoch, with the loss of 0.0874 in validation data and with the accuracy of 0.9682. The overall loss for the training data reached the value of 0.0451, and the accuracy was 0.9831. The test data showed a loss of 1.8078 and an accuracy of 0.7579.

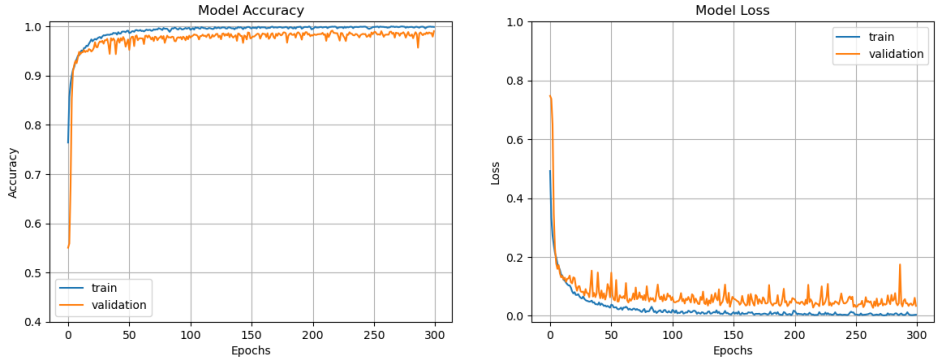
DenseNet121 The network was deprived of any pre-trained weights in order to train from scratch on the prepared data. The upper layers were removed to create



(a) Accuracy of the model in subsequent epochs **(b)** Loss of the model in subsequent epochs

Figure 4.21: Training process of the VGG19 model for training data from the SzUZG set

a new layer system consisting of a Flatten layer and then a Dense layer with 512 neurons with the ReLU activation function. The next layer was Dropout at the level 0.5 and the Batch Normalization layer. Finally, the Dense layer with one neuron with a sigmoid activation function generated the results. The total number of parameters of the presented network was 32 656 017, which made it possible to use a relatively large input image package size (batch - 64). Moreover, the model was preceded by layers performing augmentation with random image rotation and vertical and horizontal flipping.



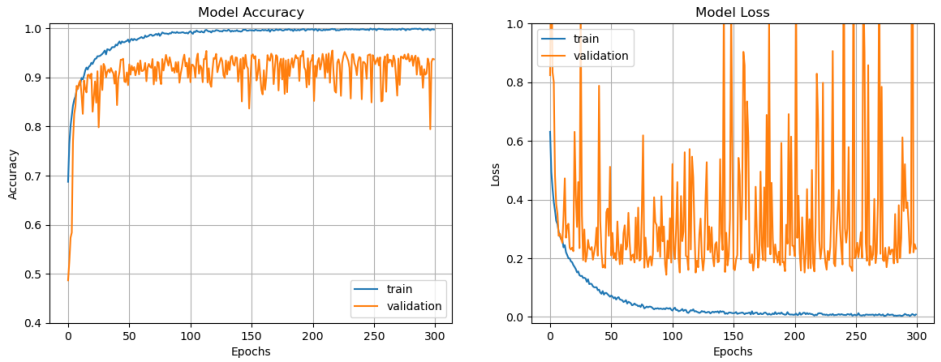
(a) Accuracy of the model in subsequent epochs **(b)** Loss of the model in subsequent epochs

Figure 4.22: Training process of the DenseNet121 model for training data from the BreakHis + GZG set

300 training epochs were allocated for training the model. The lowest result of the loss function for the validation data from the BreakHis + GZG set occurred in the 271th training epoch and amounted to 0.0271 with an accuracy of 0.9894. In the same epoch, the loss function on the training data reached 0.0020, with an

accuracy of 0.9999. Unfortunately, the result of the loss function on the test data amounted to 4.7317 with an accuracy of 0.5123.

The training on the data from the SzUZG set lasted 300 epochs, but the best result on validation data was achieved in the 120th training epoch. The value of the loss function for the validation data reached the lowest level of 0.1513 with an accuracy of 0.9394. In the same epoch, a loss function of 0.0200 and an accuracy of 0.9940 were achieved in parallel for the validation data. On the test data, a loss function result of 1.6325 and an accuracy of 0.6601 were achieved.



(a) Accuracy of the model in subsequent epochs **(b)** Loss of the model in subsequent epochs

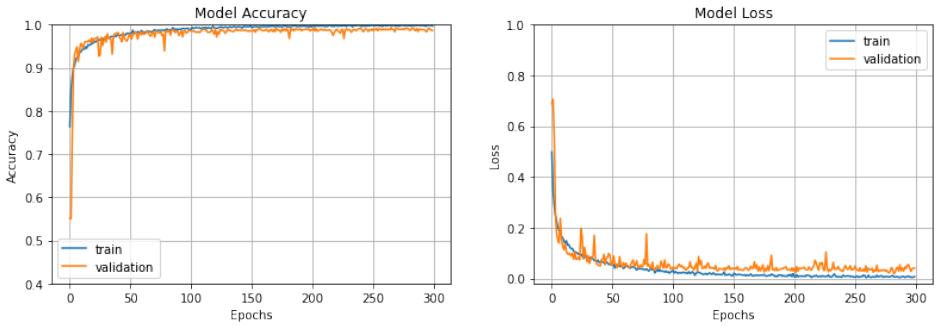
Figure 4.23: Training process of the DenseNet121 model for training data from the SzUZG set

The process of training the network on the BreakHis + GZG set observed in Fig. 4.22 is very gentle, which means that the value of the learning rate is well adjusted. The learning curve in Fig. 4.23 looks slightly worse, but thanks to the use of the same learning coefficient value, the moment of network overfitting and a characteristic trend in the validation curve after the 120th training epoch can be clearly observed.

Xception No pre-trained weights were used in the prepared base network, and the top layers were also removed. They were replaced with the following sequence: a Flatten layer, then a Dense layer with 512 neurons with a ReLU activation function, a Dropuot (0.5) layer, a Batch Normalization layer, and finally a Dense layer containing one neuron with a sigmoid activation function. Additionally, data augmentation was performed in the form of random horizontal and vertical transformations and rotations. The network was also preceded by normalization of the values z (0..255) to (-1..1). The total number of parameters of the built model reached 72 189 225, 55 552 of which were not subject to training.

The best result for the validation data from the BreakHis + GZG set was achieved in the 283rd training epoch. The error in this epoch amounted to 0.0198, with an accuracy of 0.9918, while for the training data the loss in the 283rd epoch

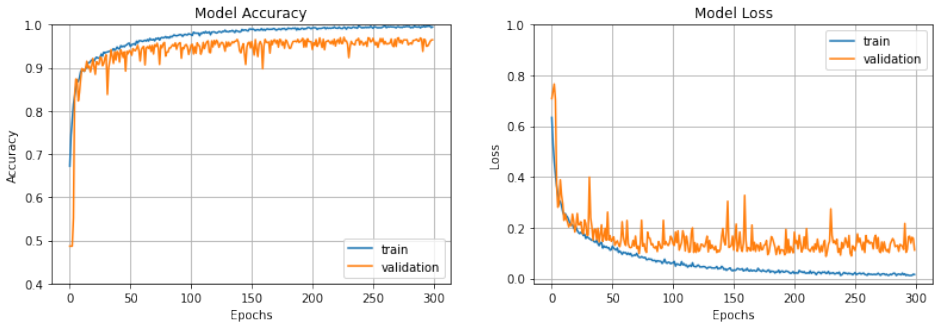
was 0.0056 and the accuracy was 0.9981. For the test data, the loss result amounted to 3.7645 with an accuracy of 0.5158.



(a) Accuracy of the model in subsequent epochs **(b)** Loss of the model in subsequent epochs

Figure 4.24: Training process of the Xception model for the BreakHis + GZG training set

On the data from the SzUZG set, the smallest loss in the validation set occurred in the 227th training epoch and amounted to 0.0869, while the accuracy reached 0.9707. The training set in the same epoch achieved a loss of 0.0195 and an ac- curacy of 0.9945, respectively. The test data, in turn, showed a loss of 1.8638 and the accuracy of the classifier was 0.6888.



(a) Accuracy of the model in subsequent epochs **(b)** Loss of the model in subsequent epochs

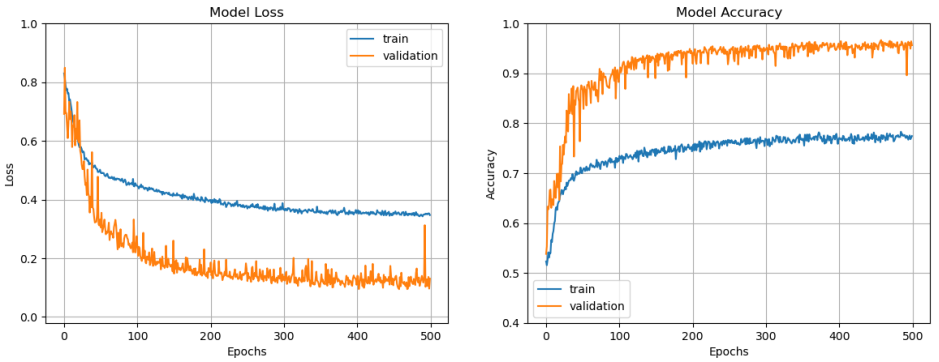
Figure 4.25: Training process of the Xception model for training data from the SzUZG set

InceptionResNetV2 The InceptionResNetV2 (Szegedy, Ioffe, Vanhoucke & Alemi 2017) network belongs to a group of residual networks. In this case, however, the features of residual networks are combined with an improved version of the Inception network (Szegedy *et al.* 2016). The reason why the authors of the model undertook experiments was to check whether the performance of the deep network

would be maintained when replacing the filter concatenation stages with residual connections (Szegedy *et al.* 2017). The use of residual connections results from the fact that the Inception network is very deep, and therefore increases the chance of gradients disappearing during training.

The network was prepared for training from scratch, i.e. the weights were randomly initialized. The top layers were removed and replaced with a set of layers starting with the Flatten layer, then the Dense layer with 512 neurons with the ReLU activation function. After this layer, a Dropout layer with a value of 0.5 was placed. The penultimate layer was Batch Normalization followed by a layer with one neuron with sigmoid activation. This model was also preceded by data augmentation layers with random image rotation and vertical and horizontal flipping. The total number of model parameters was 74 000 609, including 61 568 parameters not subject to training.

Due to the very large number of network layers, the entire training process was extended to 500 epochs. The best result on the validation data for the BreakHis + GZG set occurred in the 458th training epoch. The observed value of the loss function on the training data was 0.0948 and the accuracy was 0.9665. In the same training epoch, the result for the loss function was 0.3528 and the accuracy was 0.7758 on the training set. Interestingly, for the first time the training data obtained worse results than the validation data. Despite this anomaly, the test data did not differ from the results obtained in other network models, namely: the value of the loss function was 3.5477 with an accuracy of 0.5158.

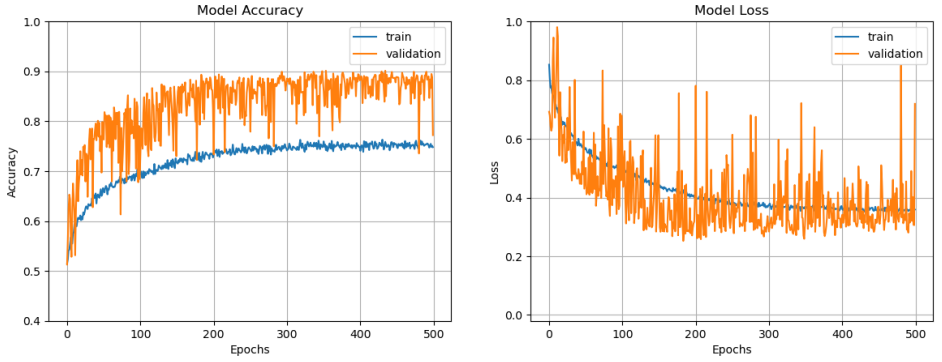


(a) Accuracy of the model in subsequent epochs **(b)** Loss of the model in subsequent epochs

Figure 4.26: Training process of the InceptionResNetV2 model for training data from the BreakHis + GZG set

On the training set of images from the SzUZG set, the best result was obtained already in the 184th training epoch. The loss value for the validation data in this epoch was 0.2522 with an accuracy of 0.8932. In the same training epoch, the loss value on the training data was 0.4552 and the accuracy was 0.7186. Also in this case, the InceptionResNetV2 network achieved better results on the validation set than on the training set. Paradoxically, the accuracy result obtained (0.7031) on

the test data was close to the result obtained on the training data. The value of the loss function on the test data reached the result of 0.8288.



(a) Accuracy of the model in subsequent epochs **(b)** Loss of the model in subsequent epochs

Figure 4.27: Training process of the InceptionResNetV2 model for training data from the SzUZG set

Both in Fig. 4.26 and 4.27 the results on the validation data achieve better results than on the training data. Moreover, for the data in Fig. 4.27, overfitting occurs very quickly, while in Fig. 4.26 it seems that there is no clear effect of overfitting, but the curves for the training and validation data are significantly separated from each other. In general, the training course on the InceptionResNetV2 network differs from the curves observed in other network models.

Results summary The tested models of deep neural networks have so far been successfully used in tasks related to multi-class classification (e.g. 1000 classes) and large data sets. In the presented experiment, we are dealing with the classification of dichotomous data on a relatively small data set. While the number of images itself oscillates around 10 000 to 12 000 depending on the collection, it should be emphasized that these images come from several dozen patients from the SzUZG collection and several dozen from the BreakHis collection. Further problems concern the quality of the data, the equipment used for scanning, and the procedures for collecting the material. Interestingly, all models except InceptionResNetV2 achieved high results on training and validation data. This means that the fit of the models to the data is equally extreme, which in turn translates into unsatisfactory results on the test data. The reasons may be attributed to the amount of available medical data and, consequently, to poor diversity of cases despite the use of augmentation. Paradoxically, the excess of input information in the form of three-channel 230×230 images may contribute to the problem. In subsequent experiments, the effect of reducing the dimensions of input images on the flexibility of selected deep neural network models will be checked.

Data from the BreakHis + GZG set were tested on the SzUZG set. The results obtained on the test set reached values from 0.5020 to 0.5242 (Tab. 4.3).

Table 4.3: Summary of image classification results in RGB space

BreakHis + GZG training data; SzUZG test data						
CNN model	**Train**		**Validation**		**Test**	
	ACC	LOSS	ACC	LOSS	ACC	LOSS
Resnet50	0.9944	0.0175	0.9914	0.0264	0.5242	3.2591
ResNet151V2	0.9916	0.0241	0.9898	0.0275	0.5151	3.6754
VGG16	0.9982	0.0063	0.9943	0.0168	0.5020	6.0449
VGG19	0.9968	0.0112	0.9918	0.0241	0.5032	6.4768
DenseNet121	0.9999	0.0020	0.9894	0.0271	0.5123	4.7317
Xception	0.9981	0.0056	0.9918	0.0198	0.5158	3.7645
InceptionResNetV2	0.7758	0.3528	0.9665	0.0948	0.5158	3.5477

SzUZG training data; BreakHis + GZG test data						
CNN model	**Train**		**Validation**		**Test**	
	ACC	LOSS	ACC	LOSS	ACC	LOSS
Resnet50	0.9767	0.0612	0.9568	0.1123	0.6003	1.8699
ResNet151V2	0.9881	0.0351	0.9657	0.1034	0.6562	1.6730
VGG16	0.9746	0.0618	0.9623	0.1035	0.7359	1.6542
VGG19	0.9831	0.0451	0.9682	0.0874	0.7579	1.8078
DenseNet121	0.9940	0.0200	0.9394	0.1513	0.6601	1.6325
Xception	0.9945	0.0195	0.9707	0.0869	0.6888	1.8638
InceptionResNetV2	0.7186	0.4252	0.8932	0.2522	0.7031	0.8288

These results mean that the obtained models have virtually no ability to generalize between data from different medical centers. The obtained models fit only the training data. The results for models trained on the SzUZG set and tested on the BreakHis + GZG set look slightly better, i.e they range from 0.6003 to 0.7579. The lowest value was achieved by the ResNet50 model and the highest by VGG19. The value obtained by the VGG19 model can be considered a good result showing the generalization ability of the model. Overall, this experiment demonstrated the tendency of deep learning methods to overfit to training data based on RGB images.

Images after normalization using H&E deconvolution

Results summary The above experiment was also modified by introducing the reduction of the dimensionality of images from three-channel space to single-channel space. This procedure was performed using image deconvolution. The deconvolution was performed in Fiji (Schindelin *et al.* 2012) software using the deconvolution matrix (H&E). The deconvolution procedure is described in section 2.3. The experiment used images in which deconvolution identified places of hematoxylin exposure. The objects in which hematoxylin accumulates are primarily cell nuclei, i.e. key diagnostic elements of histopathological images. In the experiment, most

of the models used in the previous study were selected for images in three-channel space with 8-bit depth.

Table 4.4: Summary of image classification results after deconvolution

BreakHis + GZG training data; SzUZG test data						
CNN model	**Train**		**Validation**		**Test**	
	ACC	LOSS	ACC	LOSS	ACC	LOSS
Resnet50	0.9149	0.1989	0.8654	0.3273	0.5306	2.2535
VGG19	0.9197	0.1845	0.8662	0.2973	0.5183	2.5664
DenseNet121	0.9778	0.0701	0.8658	0.3408	0.5257	2.8245
Xception	0.7587	0.4709	0.7071	0.5179	0.4848	17.9988
InceptionResNetV2	0.8709	0.2887	0.8490	0.3289	0.5385	1.8650

SzUZG training data; BreakHis + GZG test data						
CNN model	**Train**		**Validation**		**Test**	
	ACC	LOSS	ACC	LOSS	ACC	LOSS
Resnet50	0.9106	0.2156	0.8868	0.2644	0.6617	1.0817
VGG19	0.9135	0.1995	0.9086	0.2270	0.7019	1.3525
DenseNet121	0.9419	0.1494	0.9076	0.2441	0.6627	1.1087
Xception	0.8107	0.3942	0.7974	0.4246	0.4474	11.0942
InceptionResNetV2	0.9188	0.1963	0.8843	0.2616	0.6228	1.1630

The obtained results (Tab. 4.4) fail to show any improvement compared to models built on three-channel images, and in the case of the Xception network, they even significantly worsen the results. To summarize the experiment, it can be stated that reducing the dimensionality of the input image to single-channel images after deconvolution fails contribute to improving the generalization ability of the CNN network. The obtained results are similar to those obtained on images in RGB space.

Images after normalization using segmentation

Results summary The last experiment consisted in classifying input images to a normalized network using the proposed hybrid segmentation method. These images are characterized by having one channel and a binary representation of pixels in this channel. The basis for the experiment was the assumption that normalization to the binary space of images would reduce the training sets to a transparent and simple form, and only the shape of the objects would then affect the final training of the model.

The results obtained on the test data (Tab. 4.5) clearly show that the data in binary images were so close to each other that it was possible to perform effective classification. On the BreakHis + GZG training set, the obtained accuracy level was lower than on the testing set, as well as on most validation sets. In fact, this may mean that the generalization potential of the model has significantly improved

Table 4.5: Summary of image classification results after segmentation

BreakHis + GZG training data; SzUZG test data						
CNN model	**Train**		**Validation**		**Test**	
	Acc	Loss	Acc	Loss	Acc	Loss
Resnet50	0.7220	0.5393	0.7095	0.5382	0.7813	0.4539
VGG19	0.7283	0.5129	0.7536	0.4774	0.7897	0.4596
DenseNet121	0.7373	0.5202	0.7364	0.4991	0.7695	0.4977
Xception	0.7191	0.5579	0.7254	0.5269	0.7811	0.4703
InceptionResNetV2	0.7666	0.5042	0.7417	0.5077	0.7806	0.4691

SzUZG training data; BreakHis + GZG test data						
CNN model	**Train**		**Validation**		**Test**	
	Acc	Loss	Acc	Loss	Acc	Loss
Resnet50	0.8022	0.4711	0.7939	0.4788	0.7004	0.5539
VGG19	0.7761	0.4786	0.8287	0.4244	0.7239	0.5275
DenseNet121	0.8267	0.4049	0.8093	0.4378	0.7121	0.5605
Xception	0.7724	0.4746	0.8183	0.4173	0.7122	0.5242
InceptionResNetV2	0.7383	0.5159	0.8019	0.4420	0.7049	0.5813

at the level of binary data compared to images with a larger number of shades. If we averaged all the accuracy results from the training, validation and test sets in both data sets, the highest result was achieved by the VGG19 model, but these differences are very small. The key indicator will be the result achieved for the test data, and in this case the VGG19 model also turned out to be slightly better than the others. However, these differences are not significant and it can be said that all models deal with these data with similar accuracy.

4.8.3 Developed classification system

4.8.3.1 Experiment plan

The second part of the experiments was meant to examine the impact of individual superior classifiers in the developed system and to demonstrate whether the proposed method of fusion between manual and deep features would improve the results obtained for models built on separate sets of manual and deep features. This part also examined the issue of improving classification results using a set of deep feature generators in relation to the individual results obtained for the tested CNN networks. The study used 8 different classification models, which were tested in various configurations. The results contain only selected best configurations. The diagram of the experiments, which was conducted with the participation of superior classifiers based on manual, deep and fusion features, is presented in Fig. 4.28. The order in which subsequent experiments are presented is based on classifiers applied in the study. Under each paragraph, there is a comment and two tables with results. One of them concerns models built on data from the Break-

His + GZG set, which were tested on the SzUZG set. The second table concerns models built on data from the SzUZG set, which were tested on BreakHis + GZG data.

Set of manual features	Set of deep features	Fusion of deep and manual features
Regression I) Training data BreakHis+GZG, Test data SzUZG II) Training data SzUZG, Test data BreakHis+GZG	Regression I) Training data BreakHis+GZG, Test data SzUZG II) Training data SzUZG, Test data BreakHis+GZG	Regression I) Training data BreakHis+GZG, Test data SzUZG II) Training data SzUZG, Test data BreakHis+GZG
Decision tree I) Training data BreakHis+GZG, Test data SzUZG II) Training data SzUZG, Test data BreakHis+GZG	Decision tree I) Training data BreakHis+GZG, Test data SzUZG II) Training data SzUZG, Test data BreakHis+GZG	Decision tree I) Training data BreakHis+GZG, Test data SzUZG II) Training data SzUZG, Test data BreakHis+GZG
Random forest I) Training data BreakHis+GZG, Test data SzUZG II) Training data SzUZG, Test data BreakHis+GZG	Random forest I) Training data BreakHis+GZG, Test data SzUZG II) Training data SzUZG, Test data BreakHis+GZG	Random forest I) Training data BreakHis+GZG, Test data SzUZG II) Training data SzUZG, Test data BreakHis+GZG
Gradient boosting I) Training data BreakHis+GZG, Test data SzUZG II) Training data SzUZG, Test data BreakHis+GZG	Gradient boosting I) Training data BreakHis+GZG, Test data SzUZG II) Training data SzUZG, Test data BreakHis+GZG	Gradient boosting I) Training data BreakHis+GZG, Test data SzUZG II) Training data SzUZG, Test data BreakHis+GZG
Naive Bayes classifier I) Training data BreakHis+GZG, Test data SzUZG II) Training data SzUZG, Test data BreakHis+GZG	Naive Bayes classifier I) Training data BreakHis+GZG, Test data SzUZG II) Training data SzUZG, Test data BreakHis+GZG	Naive Bayes classifier I) Training data BreakHis+GZG, Test data SzUZG II) Training data SzUZG, Test data BreakHis+GZG
k-NN classifier I) Training data BreakHis+GZG, Test data SzUZG II) Training data SzUZG, Test data BreakHis+GZG	k-NN classifier I) Training data BreakHis+GZG, Test data SzUZG II) Training data SzUZG, Test data BreakHis+GZG	k-NN classifier I) Training data BreakHis+GZG, Test data SzUZG II) Training data SzUZG, Test data BreakHis+GZG
Support vector machine I) Training data BreakHis+GZG, Test data SzUZG II) Training data SzUZG, Test data BreakHis+GZG	Support vector machine I) Training data BreakHis+GZG, Test data SzUZG II) Training data SzUZG, Test data BreakHis+GZG	Support vector machine I) Training data BreakHis+GZG, Test data SzUZG II) Training data SzUZG, Test data BreakHis+GZG
Neural classifier I) Training data BreakHis+GZG, Test data SzUZG II) Training data SzUZG, Test data BreakHis+GZG	Neural classifier I) Training data BreakHis+GZG, Test data SzUZG II) Training data SzUZG, Test data BreakHis+GZG	Neural classifier I) Training data BreakHis+GZG, Test data SzUZG II) Training data SzUZG, Test data BreakHis+GZG

Figure 4.28: Scheme of experiments on master classifiers

4.8.3.2 Introduction

In the experiments, image sets were used to extract two groups of features: manual and deep. Manual features were obtained based on shapes, colors and textures of cell nuclei detected in the images by means of a segmentation network. Deep features were found in output values of individual neurons occurring in the penultimate layer of deep neural networks. In both cases, the isolated features were one-dimensional and therefore could be combined into a common set of features. This property was used to investigate whether combining manual and deep features into one set improved classification results. The classification experiments were carried out using JMP Pro (SAS Institute Inc. 1989-2022) software. Malig-

nant cases marked with a value of 1 were considered positive (TP), and benign cases marked with a value of 0 were considered negative (TN).

4.8.3.3 Imputation of missing data

At the stage of feature extraction and statistical calculations, some of the data were not calculated due to numerical problems. In the case of several features, the number of missing data did not exceed 2. One feature, namely the correlation entropy of the GLCM matrix, showed a significant number of missing data, i.e. 7154 out of all 22 326 rows obtained from the BreakHis + GZG set and the SzUZG set. Unfortunately, this number strongly affects the reliability of the feature obtained after data imputation. The imputation method used was based on replacing the missing data by means of a least squares method based on existing data and a normal distribution model.

4.8.3.4 Experiments

Regression Regression offers a wide range of available methods, dimensionality reduction, regularization and validation. Various validation methods were tested in the experiments: Bayesian information criterion, Akaike information criterion and cross-validation (KFold, Holdout). Tests of logistic regression models and regularized regression models were also carried out, including: ridge regression, Lasso regression and elastic net regression. The results for the logistic regression model with stepwise feature elimination were also examined. It is usually assumed that cross-validation is a better validation method than the Bayesian information criterion. However, more effective feature elimination and the extent of final reduction of the model dimensionality constitute an advantage of this information criterion. The differences between the tested regression models were small, so the summary will include the results of the best two methods. Only the full results for the elastic net regularization regression model validated using Bayesian information criterion are included in this section, as the stepwise regression method was unable to complete the feature selection task due to computational complexity for the manual feature set.

Table 4.6: Regression with Elastic Net regularization and validation using the Bayes information criterion, trained on the BreakHis + GZG set and tested on the SzUZG set

	Manual			**Deep**			**Fusion**		
	TPR	TNR	ACC	TPR	TNR	ACC	TPR	TNR	ACC
Train.	0.8100	0.8636	0.8396	0.8612	0.7293	0.7883	0.8627	0.8574	0.8597
Valid.	-	-	-	-	-	-	-	-	-
Test	0.4594	0.9198	**0.6967**	0.7030	0.8620	**0.7849**	0.6606	0.8961	**0.7819**

The model was calculated based on manual features from the BreakHis + GZG set. Out of the total number of 251 features, 173 features plus an intercept were added to the model after reduction, i.e. the set was reduced by approximately

30% of its original state. The AUC field reached a value of 0.9177 for the training data and a value of the generalized R^2 of 0.6358. The building of the model based on deep features concluded with the addition of an intercept and 14 out of 25 features, i.e. equal to 60% of the original number. The obtained value of the generalized coefficient of determination amounted to 0.5567, while the AUC field was estimated at 0.8762. However, the greatest reduction in dimensionality concerned the model based on the fusion between manual and deep features. The model needed 96 features plus an intercept, i.e. less than 40% of all features, obtaining the value of the generalized coefficient R^2 at the level of 0.6939, with the AUC field at 0.9360.

The summary of results for the test data (Tab. 4.6) for a classifier trained on BreakHis + GZG data shows that the worst classification result was achieved by a classifier built with a set of manual features, while the best one with a set of deep features. The classifier obtained on feature fusion achieved a result almost identical to the set of deep features.

Table 4.7: Regression with Elastic Net regularization and validation using the Bayes information criterion, trained on the SzUZG set and tested on the BreakHis + GZG set

	Manual			**Deep**			**Fusion**		
	TPR	TNR	ACC	TPR	TNR	ACC	TPR	TNR	ACC
Train.	0.8365	0.8751	0.8564	0.8568	0.8601	0.8585	0.8779	0.8849	0.8815
Valid.	-	-	-	-	-	-	-	-	-
Test	0.6418	0.8733	**0.7697**	0.7968	0.6400	**0.7102**	0.7124	0.8333	**0.7792**

As a result of regularization, the set of manual features from the SzUZG set was reduced to 125 features and an intercept. The obtained result of the generalized coefficient of determination was 0.6862. The AUC value for the classifier built on manual features was 0.9328. The set of deep features was not reduced as a result of regularization, so all features turned out to be important in the model. The value of the generalized R^2 coefficient was 0.7020, and the AUC coefficient was 0.9371. Feature fusion, in turn, was characterized by a model built on the basis of 70 features and an intercept, i.e. regularization contributed to a significant reduction in the number of features. At the same time, the value of the generalized coefficient of determination reached the result of 0.7556, while the AUC field reached the value of 0.9545.

In the case of classifiers built on a set of SzUZG training images, the model based on a set of deep features performed worst, while the model based on feature fusion performed best, slightly improving the model built on a set of manual features.

Taking into account the average accuracy values (Tab. 4.7) obtained on the basis of both data sets (BreakHis + GZG and SzUZG), the best results were achieved by models based on feature fusion. The average values were 0.7805 for fusion, 0.7475 for deep features and 0.7332 for manual features. Feature fusion therefore improves the accuracy of the model on the test data by at least 3% on average. However, taking into account the minimum values obtained for the

models: fusion (0.7792), deep (0.7102) and manual (0.6967), it can be seen how generalization properties of the model based on feature fusion improved when compared to other models.

Decision tree The sets used for training and validating the model were divided in the ratio of 0.8 : 0.2. This means that the prepared validation sets amount to approximately 2,000 observations. The selection of the best division involves finding the maximum value of the determination coefficient R^2, while the stopping criterion occurs after 10 subsequent divisions with no improvement in the result. The advantage of the decision tree (DT) model is its transparent structure, while its disadvantage is that it narrows the set of observations in subsequent divisions. The results of the experiments performed are presented in Tab.: 4.8 and 4.9. In the

Table 4.8: DT, trained on the BreakHis + GZG set, and tested on the SzUZG set

	Manual			**Deep**			**Fusion**		
	TPR	TNR	ACC	TPR	TNR	ACC	TPR	TNR	ACC
Train.	0.8718	0.8955	0.8850	0.9166	0.6779	0.7849	0.9072	0.8626	0.8825
Valid.	0.8307	0.8777	0.8559	0.9104	0.6862	0.7857	0.8827	0.8296	0.8537
Test	0.6565	0.7799	**0.7201**	0.7485	0.8156	**0.7830**	0.7104	0.8266	**0.7702**

case of the BreakHis+GZG set, the DT division for data with manual features was completed in 57 iterations. Then the highest value of the generalized coefficient R^2 was achieved for the validation set and amounted to 0.6673. In the same iteration of the training set, the value of the generalized coefficient of determination reached the value of 0.7473. The area size (AUC) was determined under the ROC curve plot and amounts to 0.9262 for the validation data and 0.9495 for the training data. Then, a DT model was performed analogously for deep features. Due to the much smaller total number of deep features, it could be expected that the number of iterations needed to generate the best model would be reduced. It is therefore not surprising that only 6 tree divisions were performed to obtain the best model. The results for this division return the value of 0.5489 for generalized R^2 for validation data and 0.5467 for training data, with AUC field for validation data at 0.8674 and for training data at 0.8648. The last test was performed on a combined set of manual and deep features. It was labeled as fusion set. Despite its considerable size, the set returned the highest result of the generalized coefficient of determination for the validation set in the 49th division and amounted to 0.6880. In the same iteration, the generalized R^2 coefficient for the training data reached the value of 0.7607. The AUC fields amounted to 0.9327 and 0.9542 for the validation and training sets, respectively.

The best classification result for the test data was achieved by a model built on a set of deep features. The worst one was based on manual features. The model built on the fusion of manual and deep features turned out to be slightly worse than the model based solely on deep features.

The highest value (0.5579) of the R^2 determination coefficient for the validation set for manual features was achieved in 24 tree splits. This division was also

Table 4.9: DT, trained on the SzUZG set, and tested on the BreakHis + GZG set

	Manual			**Deep**			**Fusion**		
	TPR	TNR	ACC	TPR	TNR	ACC	TPR	TNR	ACC
Train.	0.8198	0.8441	0.8323	0.8261	0.8802	0.8541	0.8622	0.8722	0.8674
Valid.	0.7891	0.8215	0.8059	0.8200	0.8466	0.8334	0.8370	0.8693	0.8535
Test	0.5461	0.7482	**0.6578**	0.7345	0.6797	**0.7042**	0.6372	0.7132	**0.6792**

characterized by the value of 0.6351 for the R^2 coefficient in the training set. The AUC coefficient in the training set reached the value of 0.9123, while it amounted to 0.8893 in the validation set. For the deep feature set, the highest R^2 value for the validation data was 0.6488 in 15 splits, where the R^2 value for the training data was 0.6998. For deep features, the AUC coefficient in the training set was 0.9332, while it amounted to 0.9165 in the validation set. The combined sets of manual and deep features enter the decision tree last. The tree model was selected in 16 splits. Interestingly, the first two divisions were made within deep features, and the next ones mainly within manual ones. The determination coefficient for the validation data was 0.6584, while for the training data it amounted to 0.7225. The AUC coefficient for the validation data reached the value of 0.9233 and 0.9407 for the training data.

Decision tree models built on training data from the SzUZG set yet again returned the best result for the set of deep features (0.7042), while the model built on a set of manual features achieved the worst result (0.6578). Feature fusion improved the results compared to the set of manual features, while performing significantly worse than the set of deep features.

Overall, the classification results on the test data showed that, on average, the best decision tree model was based on deep features, achieving a score of 0.7436. The other average scores were 0.7247 for feature fusion and 0.6889 for manual features. The minimum score ranking is also similar. The conclusion from the experiment is that a set of deep features turned out to be the best for decision tree models.

Random forest As the name suggests, the RF classifier requires the initial declaration of the number of trees included in this forest, or rather to be used in the forest. The default number of trees selected for the experiment was 100. Another important parameter is the number of columns (features) that will be considered for division in the next division iteration. This value is set by default at 3/4 of the total number of columns in the data set. You can also enter a bootstrap coefficient. The minimum number (10) and maximum (2000) number of splits in a single tree are also defined. Additionally, there is an option to define the minimum number of samples for which a division can be performed. This number was set at 12. The early stop option is available when using the validation set. Stopping early means concluding the training process when adding more trees fails to improve the validation result.

Table 4.10: RF, trained on the BreakHis + GZG set, and tested on the SzUZG set

	Manual			**Deep**			**Fusion**		
	TPR	TNR	ACC	TPR	TNR	ACC	TPR	TNR	ACC
Train.	0.9805	0.9761	0.9780	0.9440	0.8649	0.9000	0.9787	0.9744	0.9763
Valid.	0.8930	0.8969	0.8951	0.8565	0.7677	0.8088	0.9109	0.9000	0.9050
Test	0.6411	0.8227	**0.7347**	0.6962	0.8562	**0.7787**	0.6968	0.8601	**0.7810**

The results for the RF classifier are characterized by a significant fit to the training data (Tab. 4.10). The classifier trained on manual features from the BreakHis + GZG set obtained a value of the generalized R^2 coefficient of 0.8811 for the training data, and 0.7846 for the validation data. The ROC curve for the training data was exemplary, and the AUC reached a value of 0.9985. For the validation data, the AUC field also scored very high, i.e. 0.9697. The subsequent task was to build a model based on a set of deep features. The value of the generalized R^2 coefficient for the training data was 0.7444, while for the validation data it amounts to 0.5840. The AUC field for the training data reached 0.9749, and for the validation data it amounted to 0.8868. The last model for the BreakHis + GZG set was built on feature fusion and obtained a generalized R^2 value of 0.8892 for the training data and 0.8041 for the validation data. In turn, the AUC field for the training data was 0.9981, and 0.9731 for the validation data.

The classifier model built on a set of manual features obtained the poorest result of 0.7347. Models built on deep features and feature fusion achieved similar results of approximate 0.7800 accuracy.

Table 4.11: RF, trained on the SzUZG set, and tested on the BreakHis + GZG set

	Manual			**Deep**			**Fusion**		
	TPR	TNR	ACC	TPR	TNR	ACC	TPR	TNR	ACC
Train.	0.9646	0.9753	0.9701	0.9338	0.9263	0.9299	0.9720	0.9671	0.9695
Valid.	0.8279	0.8683	0.8488	0.8604	0.8595	0.8600	0.8804	0.8820	0.8812
Test	0.5375	0.8104	**0.6883**	0.7992	0.6329	**0.7073**	0.7073	0.7064	**0.7068**

The set of manual features obtained from the SzUZG set was used to build a model that obtained a generalized coefficient of determination value of 0.8566 for the training data. On the validation data, the generalized coefficient of determination reached 0.6868. The AUC field on the training data was 0.9968 and on the validation data it amounted to 0.9336. The second stage involved building a model on a set of deep features. This set achieved a generalized R^2 value of 0.8395 for the training data and 0.7082 for the validation data. The AUC area under the ROC curve reached 0.9859 for the training data and 0.9385 for the validation data. The last experiment involved building a model based on feature fusion. The generalized R^2 coefficient for the training data reached 0.8895, while for the validation data it was 0.7690. The AUC field reached a value of 0.9970 on the training data and 0.9595 on the validation data.

Among the models built on the SzUZG set, the worst result (Tab. 4.11) went to a classifier built on the set of manual features (0.6883). The best result was achieved by models built on a set of deep features (0.7073) and feature fusion (0.7068), the difference between them being negligible. It is worth noting that the levels of TPR and TNR are balanced in the model based on feature fusion.

The average classification results on the test sets for pairs of models were 0.7439 for fusion, 0.7430 for deep features and 0.7115 for manual features. The minimum values for individual pairs of models are consistent with the results obtained on the SzUZG set. Despite very high classification results on training and validation data, classifiers based on RF failed to confirm equally high effectiveness on test data. These models probably tend to overfit the training data.

Gradient boosting (boosted trees) The first parameter that can be determined before building a BT model is to determine the maximum number of layers (trees) that can be added. In all experiments, this value was set at 200. Another factor that was set was the number of divisions in each layer. This parameter was set at 18 divisions. The next parameter was the learning rate. Its value was set within the limits of 0 to 1, taking into account the fact that the higher its value, the faster the model convergence can be achieved, but at the same time the risk of overfitting the model will be increased (SAS Institute Inc. 1989-2022). A value of 0.05 was adopted in the experiments. The last important parameter to determine was the minimum number of observations on which the division can be made. It was set at 5. In order to maintain the repeatability of the selected sets, a random selection of the set's division into validation and training data was reset. Building a model

Table 4.12: BT, trained on the BreakHis + GZG set, and tested on the SzUZG set

	Manual			**Deep**			**Fusion**		
	TPR	TNR	ACC	TPR	TNR	ACC	TPR	TNR	ACC
Train.	0.9165	0.9309	0.9245	0.8578	0.7633	0.8052	0.9213	0.8989	0.9089
Valid.	0.8476	0.8523	0.8501	0.8601	0.7585	0.8055	0.8850	0.8615	0.8724
Test	0.6846	0.7932	**0.7405**	0.7061	0.8576	**0.7841**	0.7366	0.8439	**0.7919**

based on the BreakHis+GZG data set based on a set of manual features resulted in a generalized R^2 coefficient of 0.7399 for the training data and 0.6532 for the validation data. The AUC coefficient for the training data was 0.9824 and for the validation data it amounted to 0.9390. The model needed 135 layers to obtain the best result. The next model was built based on a set of deep features. This time, the number of layers used in the tree was 91. The value of the generalized coefficient of determination was 0.6039 for the training data and 0.5786 for the validation data. The AUC field for the training data obtained a result of 0.8995 and for the validation data it was 0.8827. The model built on feature fusion was characterized by 145 layers. The value of the generalized R^2 coefficient was 0.7798 for the training data and 0.7166 for the validation data. The AUC fields were estimated at 0.9771 and 0.9491 for training and validation data, respectively.

As a result of the tests performed, the worst classification accuracy (Tab. 4.12) was achieved by the model built on the basis of manual features (0.7405). The model based on a set of deep features achieved a significantly better result on the test data (0.7841), while the model based on feature fusion achieved the best result equal 0.7919.

Table 4.13: BT, trained on the SzUZG set, and tested on the BreakHis + GZG set

	Manual			**Deep**			**Fusion**		
	TPR	TNR	ACC	TPR	TNR	ACC	TPR	TNR	ACC
Train.	0.8374	0.9099	0.8747	0.8758	0.8756	0.8757	0.8939	0.8927	0.8932
Valid.	0.7733	0.8507	0.8134	0.8478	0.8624	0.8554	0.8678	0.8527	0.8600
Test	0.4218	0.8342	**0.6497**	0.8118	0.6341	**0.7136**	0.7558	0.6757	**0.7115**

Manual features (from the SzUZG set) were used to build a model whose value of the generalized coefficient of determination for the training set was 0.6960, and for the validation set 0.6043. The area under the ROC curve for the training data was 0.9525, and for the validation data it was 0.9055. The tree achieved the best result after 104 layers. In the case of the deep feature set, the number of layers needed to obtain the best model was 115, which is slightly more than for the manual ones. This is surprising considering the much smaller number of deep features in the model. The value of the generalized R^2 coefficient for the training data is 0.7561, and for the validation data it is 0.7017. The AUC field achieved a result of 0.9556 for the training data and 0.9363 for the validation data. The fusion of manual and deep features required 90 layers to obtain the best model. The training data obtained a coefficient of determination value of 0.7691 and the validation data of 0.7119. The AUC field was 0.9640 for the training data and 0.9420 for the validation data.

The classifier built on a set of manual features achieved a very poor result in detecting malignant cases (TPR 0.4218). The accuracy score for this model is increased to 0.6497 thanks to the efficient detection of benign images (TNR 0.0.8342). The other two classifiers achieved comparable results for the training data, i.e. ACC 0.7136 for deep features and ACC 0.7115 for feature fusion.

The final summary of the classifiers and the results obtained on both datasets (BreakHis + GZG and SzUZG) indicate that the average lowest score of 0.6951 was obtained for the set of manual features (Tab. 4.13). A significantly better average result was obtained for the set of deep features (0.7488) and for the set consisting of a fusion of manual and deep features (0.7517).

Naive Bayes classifier This classifier is characterized by the lack of preliminary model parameters to be determined. This means that the only option that can influence the classification result is to divide the data into validation and training data sets. It is also important to achieve the repeatability of the model, therefore the randomness of data division was reset and, similarly to other classifiers, the value of the separated validation part was set at 0.2 of the entire set.

Table 4.14: NB trained on the BreakHis + GZG set, and tested on the SzUZG set

	Manual			**Deep**			**Fusion**		
	TPR	TNR	ACC	TPR	TNR	ACC	TPR	TNR	ACC
Train.	0.6504	0.7370	0.6983	0.9016	0.6617	0.7692	0.7925	0.7093	0.7465
Valid.	0.6387	0.7391	0.6942	0.8993	0.6519	0.7618	0.8246	0.6874	0.7489
Test	0.7523	0.7776	**0.7651**	0.7483	0.8377	**0.7944**	0.7661	0.8111	**0.7893**

The NB classifier built on manual features obtained from the BreakHis + GZG set obtained an AUC results of 0.7264 (benign) and 0.7265 (malignant) for the training data and 0.7231 (benign) and 0.7224 (malignant)for the validation data. The use of deep features to build the classifier resulted in an AUC value of 0.8162 (benign and malignant) for the training data, and 0.8163 (benign) and 0.8170 (malignant) for the validation data. The classifier built on feature fusion had an AUC value for training data of 0.7777 (benign) and 0.7786 (malignant). The AUC value for the validation data was 0.7775 for both malignant and benign cases.

The NB classifier built on a set of manual features obtained the lowest ACC score (0.7523). The remaining feature sets scored 0.7944 for deep features and 0.7893 for feature fusion, respectively (Tab. 4.14). What seems interesting is the fact that the ACC accuracy coefficient obtained higher values on the test data than on the training and validation data. It can therefore be assumed that the NB classifier is highly resistant to over-fitting to the examined data.

Table 4.15: NB, trained on the SzUZG set, and tested on the BreakHis + GZG set

	Manual			**Deep**			**Fusion**		
	TPR	TNR	ACC	TPR	TNR	ACC	TPR	TNR	ACC
Train.	0.7659	0.7823	0.7743	0.8193	0.8531	0.8366	0.8148	0.8210	0.8180
Valid.	0.7881	0.7794	0.7835	0.8069	0.8571	0.8333	0.7805	0.8207	0.8011
Test	0.6028	0.7550	**0.6869**	0.8421	0.6351	**0.7277**	0.7531	0.7029	**0.7253**

The classifier built on a set of manual features obtained from the SzUZG set has AUC field values of 0.8013 (benign) and 0.7997 (malignant) for the training data. For the validation data, the AUC field sizes are 0.8063 for malignant and 0.8067 for benign cases. Another NB classifier, this time built on deep features, was characterized by an AUC field of 0.8756 (benign) and 0.8757 (malignant) for the training set. For the validation data, the AUC was 0.8681 for benign cases and 0.8680 for malignant cases. Feature fusion did not improve the AUC field values. For the training data, the value of the AUC field was 0.8337 and 0.8334 for benign and malignant cases, respectively. For the validation data, the AUC field value was 0.8174 for both benign and malignant cases.

Classifiers built on the SzUZG set are characterized by higher ACC values for training and validation data compared to test data. The lowest classification result was obtained for the model built on the basis of manual features (0.6869). Better

results were achieved by classifiers trained on a set of deep features (0.7277) and feature fusion (0.7253). It can also be seen that feature fusion causes the TPR and TNR levels to be equalized relative to their levels obtained for the set of manual and deep features.

The average highest accuracy result for both data sets was achieved by classifiers built on a set of deep features (0.7610). Models based on feature fusion achieved a slightly lower average result (0.7573). The lowest result was once again recorded for manual features (0.7260). Relatively low accuracy values obtained on the training and validation sets are a specific feature of NB (Tab. 4.15).

k-NN classifier Compared to other classifiers, this classifier has a relatively small set of parameters to be determined. Basically, the only parameter for models built on continuous variables is the maximum number of k-nearest neighbors that will be tested on the model. From this number, the one whose model achieves the lowest classification error for the validation data is selected.

Table 4.16: k-NN classifier, trained on the BreakHis + GZG set, and tested on the SzUZG set

	Manual			**Deep**			**Fusion**		
	TPR	TNR	ACC	TPR	TNR	ACC	TPR	TNR	ACC
Train.	0.7741	0.8295	0.8049	0.8714	0.7069	0.7798	0.8179	0.8224	0.8204
Valid.	0.7807	0.8546	0.8204	0.8824	0.7362	0.8039	0.8209	0.8531	0.8382
Test	0.7026	0.8318	**0.7691**	0.7333	0.8551	**0.7960**	0.6925	0.8638	**0.7808**

The first k-NN classifier model was built on manual features. This model was chosen for k of 19 neighbors because it achieved the lowest classification error for the validation data. The value of the R^2 coefficient for the training data was 0.3900, and for the validation data it amounted to 0.4102. A set of deep features was achieved by the best model for the number k with a value of 94. The obtained value of the R^2 coefficient for the training data was 0.3693, and for the validation data it amounted to 0.4093. The model built on feature fusion achieved the best classification result for the validation data for 11 neighbors. The value of the coefficient of determination for the training data was 0.4455 and for the validation data it amounted to 0.4670.

The k-NN classifiers achieved the highest result (Tab. 4.16) on the set of deep features (0.7960), followed by feature fusion (0.7808) and manual features (0.7691). Overall, it can be said that the classifier achieved high results on all feature sets. The k-NN classifier achieved the highest classification result of models tested so far on a set of manual features.

The set of manual features achieved the best classification result for the validation data for a classifier with a k value of 19. The value of the R^2 coefficient for the training data was 0.4576, and for the validation data it was 0.4669. For the set of deep features, the classifier for the validation data achieved the lowest classification error for the model built on 47 nearest neighbors. The R^2 coefficient value for the validation data was 0.5359, while for the training data it was 0.5155.

Table 4.17: k-NN classifier, trained on the SzUZG set, and tested on the BreakHis + GZG set

	Manual			**Deep**			**Fusion**		
	TPR	TNR	ACC	TPR	TNR	ACC	TPR	TNR	ACC
Train.	0.7865	0.8646	0.8266	0.8455	0.8564	0.8511	0.8425	0.8797	0.8616
Valid.	0.7775	0.8839	0.8327	0.8489	0.8654	0.8574	0.8384	0.8868	0.8635
Test	0.4539	0.8165	**0.6543**	0.8080	0.6360	**0.7129**	0.6794	0.7166	**0.7000**

The last test looked at the results for feature fusion. In this case, the optimal number of nearest neighbors was also 23, whereas the coefficient of determination for the training data was 0.5399 and for the validation data it was 0.5555.

Unlike k-NN classifiers built on the BreakHis + GZG set, models built on the SzUZG set are characterized by much lower classification results on test data (Tab. 4.17). For the set of manual features, the accuracy only amounted to 0.6543. This result indirectly results from the very low value of the TPR coefficient, which amounted to 0.4539. The remaining two models obtained the following results: 0.7129 for the deep feature set and 0.7000 for feature fusion.

The best average result, i.e. 0.7544, was achieved by the model based on deep features, a slightly lower result, i.e. 0.7404, was achieved by the model based on feature fusion, and the lowest average result, i.e. 0.7117, was achieved by models based on a set of manual features.

Support vector machine with radial activation function The SVM classifier in the JMP software is available in linear and radial versions. After a short analysis of the test results, slightly better results were obtained in radial SVM classifier models. Additionally, radial versions have proven to be less computationally complex. The radial version of the SVM classifier has two main parameters in the JMP software. The first one is Cost, which is responsible for the margin width (the lower the parameter value, the wider the margin). The Cost parameter ranges from 0 to 1, and the default value is set to 1 and left as such. A value of 1 means that the algorithm is inclined to achieve a lower classification error (SAS Institute Inc. 1989-2022). Unlike the Gamma parameter, the Cost parameter also appears in the linear version of the algorithm, which appears only in the radial version. The Gamma parameter is responsible for the curvature of the decision edge. The higher it is, the greater the curvature (SAS Institute Inc. 1989-2022). A large curvature value allows for a more flexible fit, which in turn can lead to easy overfitting (SAS Institute Inc. 1989-2022). Due to the difficulty of determining the ideal parameter, a default value of 1 divided by the number of parameters was left.

The support vector machine classifier model built on the BreakHis + GZG set (a set of manual features) achieved a value of the generalized R^2 coefficient of 0.7531 for the validation data and 0.8211 for the training data. This model was built based on a set of manual features. The total number of support vectors in this model was 4535. The AUC field for this model obtained a value of 0.9717 for the training data and 0.9547 for the validation data. The next model was built

Table 4.18: SVM with a radial activation function, trained on the BreakHis + GZG set, and tested on the SzUZG set

	Manual			**Deep**			**Fusion**		
	TPR	TNR	ACC	TPR	TNR	ACC	TPR	TNR	ACC
Train.	0.9074	0.9299	0.9198	0.8860	0.7043	0.7856	0.9298	0.9273	0.9284
Valid.	0.8706	0.8936	0.8833	0.8978	0.7262	0.8029	0.9006	0.8877	0.8935
Test	0.6618	0.8697	**0.7689**	0.7321	0.8458	**0.7907**	0.6925	0.8769	**0.7875**

on a set of deep features. This model used 4599 support vectors, while the value of the generalized coefficient R^2 was 0.4945 for the training data and 0.5222 for the validation data. The AUC fields obtained lower values than in the previous model and amounted to 0.8666 for the training data and 0.8702 for the validation data. The third model was built on a set of feature fusions and was characterized by 4088 support vectors. The value of the generalized R^2 coefficient was 0.8463 for the training data and 0.7804 for the validation data. The area under the ROC curve was 0.9780 for the training data and 0.9623 for the validation data.

Among the classifiers built on the BreakHis + GZG set and tested on SzUZG, the best classification result (Tab. 4.18) was obtained for the set of deep features (0.7907) and a slightly worse result was recorded for feature fusion (0.7875). The lowest result was obtained for the set of manual features, even if this is one of the highest results obtained for this set (0.7689).

Table 4.19: SVM with a radial activation function, trained on the SzUZG set and tested on the BreakHis + GZG set

	Manual			**Deep**			**Fusion**		
	TPR	TNR	ACC	TPR	TNR	ACC	TPR	TNR	ACC
Train.	0.8664	0.9205	0.8943	0.8541	0.8591	0.8566	0.9132	0.9162	0.9147
Valid.	0.8371	0.8815	0.8600	0.8514	0.8642	0.8580	0.8975	0.8940	0.8957
Test	0.5667	0.8504	**0.7235**	0.8163	0.6350	**0.7161**	0.7638	0.7631	**0.7634**

The next set of models will be built on the set of images from the SzUZG collection. The set of manual features was used to build the first model, which had 3534 support vectors. For this model, the value of the generalized R^2 coefficient on the training data was 0.7795, and on the validation data it was 0.6915. The AUC field for the training data was 0.9610, and for the validation data it was 0.9361. The next model was based on a set of deep features and obtained 2651 support vectors. This model, in turn, was characterized by a value of the generalized R^2 coefficient of 0.6329 for the training data and 0.6426 for the validation data. The area under the ROC curve was 0.9146 (training data) and 0.9177 (validation data). The last model was built on a set of feature fusions, and 2897 support vectors were used to build it. The generalized coefficient R^2 obtained a value of 0.8260 for the training data and 0.7800 for the validation data. The value of the AUC field for the training data was 0.9731, and for the validation data it amounted to 0.9616.

For SVM classifiers built on the SzUZG set, the highest accuracy result (Tab. 4.19) was obtained for feature fusion (0.7634). The second highest result was exceptionally obtained for the set of manual features (0.7235), while the lowest accuracy score this time was obtained for the set of deep features (0.7161).

After averaging the obtained accuracy values obtained on the test data, the ranking of the datasets is as follows: the best result (0.7754) for feature fusion, the next result is 0.7534 for the deep feature set and 0.7462 for the manual feature set. This is the highest average classification result obtained for the manual feature sets.

Neural classifier The neural classifier (NC) available in the JMP Pro (SAS Institute Inc. 1989-2022) software allows the use of a multi-layer perceptron with a maximum number of two hidden layers with the ability to define neurons according to their activation functions: hyperbolic tangent, linear and Gaussian. After many experiments, it was found that expanding the layers of hidden networks improves the quality of the classifier for training and validation data, but unfortunately worsens the results for test data. This state of affairs is most likely caused by the increase in the tendency of more extensive neural networks to overfit. Due to the above, it was decided to introduce the simplest model with one neuron with a hyperbolic tangent activation function. In addition to the above-mentioned possibilities, the NC module enables the creation of a boosted model, operating on a principle similar to BT. It is also possible to transform continuous variables into a state approximate to a normal distribution by means of the S_U or S_B Johnson distribution. Unfortunately, none of the above-mentioned modifications improves or even worsens the results for the test data. However, the results for the training and validation data do improve. The quadratic penalty function method was chosen for the models and the number of models for comparison was set at 100. This means that the best model out of 100 generated, i.e. one with the lowest error for the validation data, was selected. Five-fold cross-validation (Kfold) was chosen as the validation method, which increases the model training time compared to the simple division method (Holdout). However, when training a very simple NC model, this time becomes lower in both cases.

Table 4.20: NC with a single neuron with TanH activation, trained on the zBreakHis + GZG set, and tested on the SzUZG set

	Manual			**Deep**			**Fusion**		
	TPR	TNR	ACC	TPR	TNR	ACC	TPR	TNR	ACC
Train.	0.8409	0.8869	0.8663	0.8623	0.7346	0.7917	0.8833	0.8780	0.8804
Valid.	0.8387	0.8870	0.8654	0.8596	0.7371	0.7919	0.8961	0.8811	0.8878
Test	0.4326	0.9289	**0.6883**	0.7100	0.8558	**0.7851**	0.5793	0.9002	**0.7446**

The first NC model was built based on manual features extracted from the BreakHis + GZG set. This model was characterized by a generalized R^2 coefficient value of 0.6909 for training data and 0.6992 for validation data, while the AUC coefficient value was estimated at 0.9374 for the training data and 0.9380 for the

validation data. The next NC model was built on the basis of acquired deep features. The generalized R^2 coefficient value achieved by this model amounted to 0.5667 for the training data and 0.5416 for the validation data. The area under the ROC curve for the training data was 0.8800, and for the validation data it was 0.8700. The last model was built on feature fusion and achieved a determination R^2 coefficient value of 0.7490 for the training data and 0.7668 for the validation data. The AUC fields achieved scores of 0.9522 and 0.9575 for training and validation data, respectively.

Taking into account the set of manual features, an average accuracy result of 0.6883 was achieved, and the ratio of TPR to TNR coefficients of 0.4326 to 0.9289 is noteworthy (Tab. 4.20). This result indicates excellent classification of benign cases and poor classification of malignant cases. The best classification accuracy result of 0.7851 was obtained for the deep feature set, while a significantly worse result of 0.7446 was obtained for feature fusion.

Table 4.21: NC with a single neuron with TanH activation, trained on the SzUZG set, and tested on the BreakHis + GZG set

	Manual			**Deep**			**Fusion**		
	TPR	TNR	ACC	TPR	TNR	ACC	TPR	TNR	ACC
Train.	0.8581	0.8728	0.8657	0.8571	0.8579	0.8575	0.8925	0.9017	0.8972
Valid.	0.8567	0.8872	0.8724	0.8648	0.8651	0.8649	0.8934	0.9085	0.9012
Test	0.6890	0.8296	**0.7667**	0.7963	0.6390	**0.7093**	0.7618	0.7862	**0.7753**

NC built on manual features from the SzUZG set was characterized by a generalized R^2 coefficient value of 0.7197 for the training data and 0.7291 for the validation data. The AUC field for the training data was 0.9431, and for the validation data it was 0.9458. The model built on a set of deep features obtained a generalized determination coefficient value of 0.7014 for the training data, and 0.7043 for the validation data. The area under the ROC curve was 0.9369 for the training data and 0.9376 for the validation data. The last model was built on the basis of a fusion of manual and deep features. This model was characterized by a generalized R^2 coefficient value of 0.7934 for the training data and 0.8056 for the validation data. The value of the AUC field for the training data was 0.9647, and 0.9680 for the validation data.

For NCs built on the SzUZG set, the highest classification accuracy result of 0.7753 was obtained for feature fusion, the second result of 0.7667 was obtained for the set of manual features, and the set of deep features scored lowest at 0.7093. Interestingly, the order of results in this case was identical to that for SVM classifiers modeled on the SzUZG set.

Average accuracy values in the tests (Tab. 4.21) amounted to 0.7275 for manual features, 0.7472 for deep features and 0.7599 for fusion between manual and deep features. Feature fusion in this case increased the average accuracy of the model built on the deep feature set by more than 1%. What is puzzling is a very low classification accuracy of 0.6883 in the case of the model trained for the set of manual features on BreakHis + GZG data, and tested on SzUZG data.

Taking into account the TPR coefficient value of 0.4326, it can be concluded that low detection of images with malignant cases constituted the main problem of this classifier when based on manual data.

Summary of classification results The best classification result on the test sets was obtained for the generalized regression model with elastic net regularization. The average accuracy values for the classification of images from two different data sets for the best models based on feature fusion were: 0.7805 for regression with elastic net regularization (Fusion_se) and 0.7826 for the stepwise regression model (Fusion_rkw). Models built on the proposed feature fusion outperformed the best result of 0.7610 obtained for deep features based on the NB classifier by approximately 2%. At the same time, the result obtained for deep features was slightly better than the result obtained from the classification with the VGG19 neural network (0.7568). The best average classification result of 0.7462 for manual features was obtained for the SVM model and this result was approximately 3.5% lower than the best result for feature fusion.

It is also possible to examine and compare the lowest scores among the best averages, where the lowest result for feature fusion was 0.7792 and for deep features it was 0.7277, which was just over 5% worse, and for manual features it amounted to 0.7235, which was just over 5.5% worse than the result for feature fusion. Tab. 4.22 provides a full picture of the best results. The end of the table also features the results of image classification by means of the proposed feature selection method with L2 regularization (Fusion_ssg). The proposed feature selection method made it possible to obtain an average accuracy of image classification on two data sets at 1. The table also reveals an interesting regularity related to the classifier selected for the set of deep features. Namely, this classifier (NB) coped worst with classification on the training data. Therefore, one could alternatively use the kNN or SVM classifier, which obtained slightly worse average classification results for the test data, while returning significantly better classification results on the training data. Additionally, it can be seen that the best classifier model built on a set of deep features achieved higher classification results than the best single neural network, i.e. VGG19. The result of 0.7610 therefore confirms the slightly higher effectiveness of the proposed set of feature generators compared to single models (0.7408-0.7568).

4.8.3.5 Cancer diagnostic results based on classified test images for a single patient

Introduction A single image used for classification has a size of 230 × 230 pixels, which may turn out to be too small a resource to make a correct diagnosis. However, these small diagnostic areas are fragments of larger images, the set of which is ultimately assigned to a single patient. The final test for the group of the best classifiers was to examine the Sensitivity-Precision curve and to calculate the average precision value on its basis. This ratio will indicate which classifier performed best in classifying cancer cases at a single patient level.

Table 4.22: Summary of the best classifiers based on accuracies obtained for test data divided into feature sets (Notation BreakHis + GZG shortened to BreakHis in the table)

Data	BreakHis	SzUZG	Average
Manual features (SVM)			
Training	0.9198	0.8943	0.9070
Validation	0.8833	0.8600	0.8716
Test	0.7689	0.7235	0.7462
Deep features (NB)			
Training	0.7692	0.8366	0.8029
Validation	0.7618	0.8333	0.7975
Test	0.7944	0.7277	0.7610
Fusion_se (Elastic net)			
Training	0.8579	0.8815	0.8697
Validation	-	-	-
Test	0.7819	0.7792	0.7805
Fusion_sr (stepwise regression)			
Training	0.8574	0.8853	0.8714
Validation	-	-	-
Test	0.7726	0.7927	0.7826
Fusion_sf (stochastic feature selection)			
Training	0.8319	0.8696	0.8508
Validation	0.8295	0.8729	0.8512
Test	0.8049	0.7791	0.7920

Classification results for the model trained on the BreakHis + GZG set and tested on the SzUZG set In the case of the effectiveness of the classifier built on the BreakHis + GZG set, it can be noted that regardless of the set of features used, a very high classification accuracy result should be expected (Fig. 4.29) on the SzUZG test set. The high classification result may be rooted in a greater diversity of cases available in the training set, as well as in the fact that this set was supplemented by a set of GZGs from the same center as the testing set. On the other hand, the training set (BreakHis + GZG) is much richer in terms of individual cases than the SzUZG test set.

Classification results for the model trained on the SzUZG set and tested on the BreakHis + GZG set As established above, the SzUZG set is a poorer set in terms of individual cases, hence the conclusion that training on the SzUZG set and testing on the BreakHis + GZG set was the more difficult classification task for the prepared models. In this task, the highest classification efficiency (Fig. 4.30) was achieved for the classifier model built on a set containing a fusion of manual and deep features. The course of the sensitivity-precision curve also indicates that the model built on a set of feature fusions is the best and has the greatest generalization abilities. This curve shows better sensitivity to precision ratios in the high value range. The

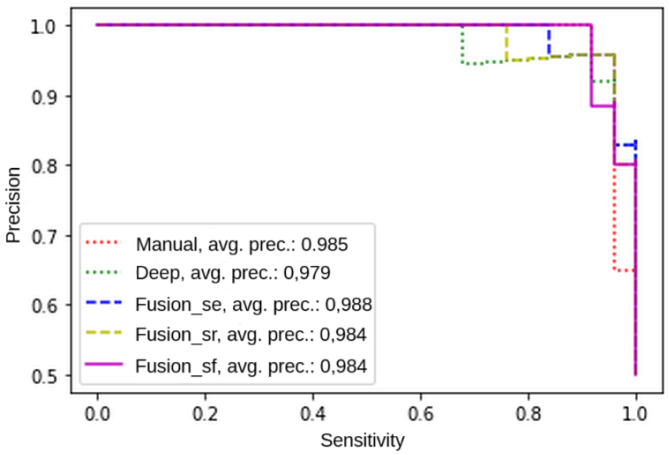


Figure 4.29: Sensitivity and precision graph for the best models trained on the BreakHis+ GZG set and tested on SzUZG

highest average precision value was obtained for a classifier built by means of the proposed feature selection method, while exceeding the value of 0.9. The highest patient classification result achieved exceeded 84%.

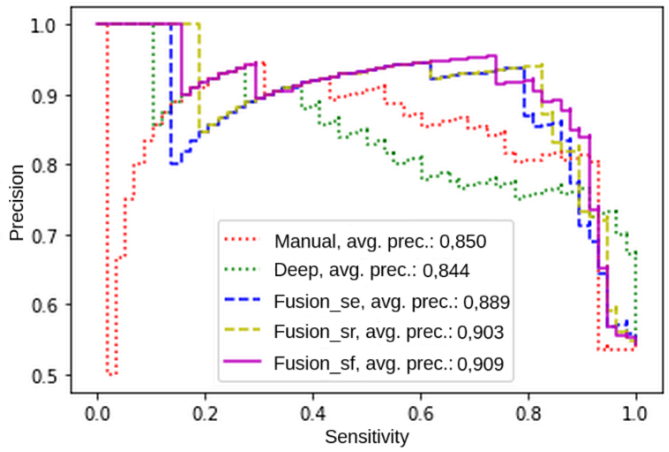


Figure 4.30: Sensitivity and precision graph for the best models trained on the SzUZG set and tested on BreakHis + GZG

Classification results for the model trained on the SzUZG set and tested on the BreakHis set The last result (fig. 4.31)is interesting from the point of view of generalization potential of the proposed classification models. The training data comes entirely

from a different medical center than the testing data. Interestingly, the maximum results obtained for all selected classifiers exceed 80%. The course of the precision-

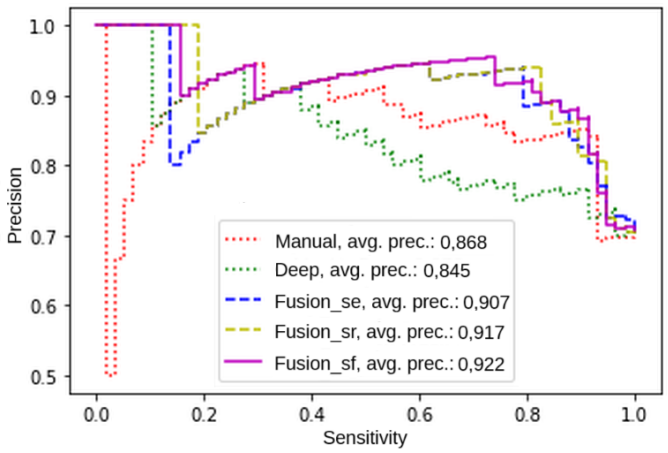


Figure 4.31: Sensitivity and precision graph for the best models trained on the SzUZG set and tested on BreakHis

sensitivity curve and the calculated average precision coefficient indicate that the best classifier was created on the basis of a fusion of features selected by means of the proposed feature selection method. Among the tested classifiers, three results exceeded the average precision value of 0.9, e.g. the logistic regression model with stepwise feature selection of 0.917 and the proposed logistic regression model with stochastic feature selection with the highest result of **0.922**.

4.9 Discussion

In the literature, it is possible to find medical image classification results reaching almost 100% for materials obtained from one medical laboratory. When taking into account classification accuracy results obtained on training and validation data from one center (SzUZG) or even slightly mixed (BreakHis + GZG), they indicate an almost perfect classification of cancer cases. For example, for the BreakHis +GZG set, the classification accuracy of the VGG16 network on the training data was 0.9982, while on the validation data it was 0.9943. Such a set of medical data, despite a certain diversity of cases, turns out to be insufficient to build a model with generalization abilities, because this model achieved a value of 0.5020 on the SzUZG test data. This problem could most likely be reduced if medical data featured huge collections of various cases. Unfortunately, in practice, medical collections usually contain several dozen cases. Another problem is the biological diversity of the cases, which means that even a doctor with many years of experience is often unsure of his diagnoses based on histopathological or cytological images, and in order to establish a diagnosis is therefore forced to perform

additional tests and collect information. Also, a large number of parameters constitutes a problem in the case of deep neural networks. These models, despite using various techniques to avoid overfitting, such as: dropout, normalization, pooling layers, regularization, and the use of validation data, are still characterized by a high tendency to overfit. A new approach to reduce the overfitting problem was to normalize input images by means of a hybrid segmentation method. Images reduced to binary masks turned out to be suitable tools for transferring knowledge between data from different medical laboratories. Interestingly, even image normalization using H&E deconvolution failed to allow the deep CNN models to achieve higher generalization abilities, and even a slight decrease was observed. This is unexpected considering that deconvolved images can be seen as an intermediate step between RGB space and binary images.

Master classifiers are characterized by various classification properties, but usually the higher the classification accuracy value they achieved on the training data, the lower the value they achieved for the test data. The RF master classifier model trained on the SzUZG set is a very good example of this phenomenon. The model was built on a set of manual features and achieved a classification accuracy result of 0.9701, while the result on test data from the BreakHis + GZG set was 0.6883. The opposite situation can be noted in the case of the NB classifier trained on the BreakHis + GZG set, where the classification accuracy of the parent classifier model built on manual features was 0.6983, while the classification accuracy on the test set was 0.7651. It can also be noticed that when the superior classifier (e.g. k-NN) achieves high classification results on the BreakHis + GZG set for test data, it achieves much worse classification results for models trained on the SzUZG set. The opposite situation occurs for methods based on decision trees. The situation is similar for regression models built on a set of manual and deep features, with the exception of models built on feature fusion. The conclusion resulting from this analysis is that the combination of the proposed feature fusion method with regression models is able to generate superior classifiers with the highest generalization potential. No other configuration achieved similar results.

The regression model based on the developed stochastic feature selection method obtained the highest average classification accuracy result. However, when broken down into individual components (Tab. 4.22), it turned out that the best result of the model trained on SzUZG data and tested on BreakHis + GZG data was achieved for stepwise regression. So why does the diagnostic result at the patient level (Fig. 4.30 and Fig. 4.31) indicate the proposed regression method with stochastic feature selection as the best model? It turns out that the stepwise regression model is poor at recognizing malignant images, but it makes up for its shortcomings in recognizing benign cases. This, in turn, translates into worse outcomes at the patient level. The regression model with stochastic feature selection distributes errors more evenly, thus gaining a diagnostic advantage over other models at the patient level.

4.10 Summary

The research conducted in this experiment made it possible for us to detect the problem of overfitting deep CNN models to training data in the form of RGB images. The use of images normalized by means of the segmentation method in the function of training data provided a solution to the encountered problem. Moreover, solving the problem of overfitting led to the construction of deep feature generators as well as to the fusion of manual and deep features. Models built on the proposed feature fusion return higher classification results and have greater generalization potential than models built solely on manual or deep features. To build the best model, regression was used with the proposed stochastic feature selection based on L2 regularization and randomization of the number of features based on gamma distribution. The regression model with stochastic feature selection demonstrated not only the best average classification accuracy at the image level, but also the highest average precision score at the patient level. As part of the experiments, additional tests related to dimensionality reduction and transfer learning were carried out, but these tests failed to demonstrate higher accuracy. The results of these studies are briefly described in Appendix A.

Chapter 5

SUMMARY

In the performed experiments, segmentation of cell nuclei in cytological images was verified using morphological methods and artificial CNN networks. As a result of research carried out on sets from one research center, it turned out that CNN methods detected cell nuclei with higher accuracy than other approaches to the segmentation problem. Deep neural networks are the most popular group of segmentation methods in the latest literature. However, the use of simpler segmentation models in some medical imaging examples may be sufficiently effective.

Classification results based on feature extraction, feature selection and the use of the most popular classifiers make it possible to achieve results of 78%-80% accuracy for data from various medical centers. Experiments with image classification using CNNs indicate an efficiency exceeding 90% only on data from one medical center. When classifying images from various medical centers, it was necessary to develop a comprehensive classification system which included the processes of segmentation, generation of deep features, extraction of manual features, fusion of deep and manual features, development of a feature selection method and selection of a superior classifier implementing the final modeling process. Effective segmentation is important for accurate classification. The segmented images were used to extract manual features and also tested as inputs to classification networks. Based on the results obtained, the best classifiers achieved 20-21% incorrectly classified individual images, which translated into 86%-96% correctly diagnosed patients.

5.1 Conclusions

The best results are achieved by the proposed system (Fig.1.1) in the configuration with the regression model as a master classifier with stochastic feature selection developed as part of the dissertation. Similar accuracy results are generated by the logistic regression model as a superior classifier with forward stepwise feature selection, but the latter model is less efficient at classifying malignant observations, while making up for its deficiency by more efficient classification of benign observations. Despite the use of various feature selection methods and superior classifiers, classification accuracy results range from 70-80%. Methods based on decision trees are the weakest in the entire list of superior classifiers.

Using the proposed approach, high accuracy of breast cancer classification (96%) was achieved based on cytological and histopathological images from various medical laboratories. The achieved accuracy is comparable to approaches in

which training and testing were performed on images from the same medical laboratory. The developed segmentation method allows for accurate segmentation of cell nuclei in cytological and histopathological images. It has been shown that the fusion of deep and manual/expert features makes it possible to increase the accuracy of breast cancer classification. It has been shown that knowledge transfer in classification tasks between cytological and histopathological images is possible.

5.2 Analysis of results and contribution to the development of the discipline

A small set of training data is sufficient for effective segmentation of cytological and histopathological images using the developed hybrid method, even for images from different medical centers that make use of different methods of material collection. The applied approach proved that with appropriate reduction of data dimensionality, it is possible to obtain high classification and diagnostic results on data obtained by means of various methods from various medical centers. Convolutional neural classifier models built on the basis of binary images obtained after segmentation of cell nuclei revealed a much higher degree of generalization and are therefore less sensitive to overfitting than models trained on images in the RGB space and after H&E normalization. The fusion of deep and manual features contributes to an overall improvement of single image classification results by 3%. Deep features obtained from one classification network are slightly inferior to a set of deep feature generators. The obtained results indicate that it is worth examining the topic of deep feature generator sets in more detail in the future. The list of the most important tasks is:

- Development of a comprehensive strategy to improve the accuracy of classification of cytological and histopathological images from various medical laboratories.
- Development of a method for segmenting instances of cell nuclei in cytological and histopathological images using the U-Net deep neural network and the watershed method.
- Development of a method for fusion of learned deep features and expert/manual features for breast cancer classification using cytological and histopathological images.
- Development of a method for extracting expert/manual features of cell nuclei from cytological and histopathological images.
- Conducting comprehensive research to verify the effectiveness of deep networks with various architectures in the task of generating deep features.
- Conducting comprehensive research to verify the effectiveness of various dimensionality reduction methods for the problem of breast cancer classification using cytological and histopathological images.
- Development of a new feature selection method for sets consisting of a fusion between manual and deep features.

5.3 Further work

Despite the high effectiveness of the CNN network, the issue of semantic segmentation of medical images requires further refinement. The problem of regulating the shapes of cell nuclei in terms of determining the boundaries of areas where objects overlap, as well as the best normalization of the input data, seem to be important for further improvement of segmentation results. Future work should also focus on such issues as:

- Constructing new features based on domain knowledge and using heuristic methods.
- Extracting new features of objects from artificial neural networks.
- Full automation of segmentation and classification of cytological images.
- Closer examination of the generalization potential in networks trained on images after segmentation.
- Obtaining new data to test the classification effectiveness of the proposed system.
- Use of serially connected U-Net models to improve segmentation of cell nuclei.
- Development of new methods for standardizing cytological and histopathological images.
- Expansion of the dataset with cytological and histopathological images with images from different medical laboratories/medical centers.

Appendix A

RESULTS OF ADDITIONAL EXPERIMENTS

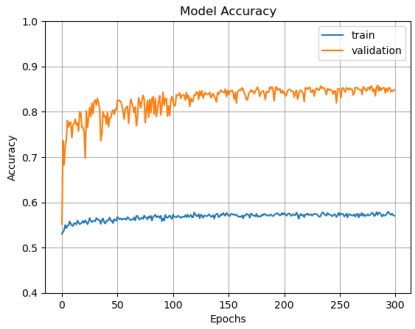
Transfer learning

Large models of deep neural networks need huge data sets to fully demonstrate their classification potential. Very often, however, the data at our disposal fails to provide such large sets of information. In addition to data augmentation used in the experiments presented above, transfer learning is one of methods of dealing with this problem. It is divided into two parts. The first one involves training only the upper layer of the network, while all layers of the main model are frozen with weights obtained on the basis of the Imagenet (Deng *et al.* 2009) set. The second part of the experiment involves the fine-tuning of the model, which involves unfreezing all layers of the network while reducing the training step.

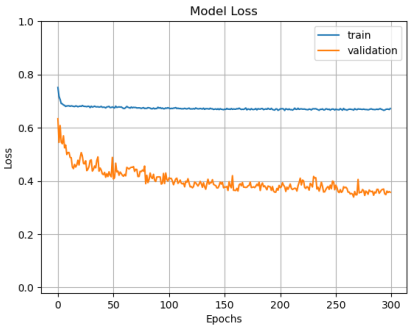
When training the InceptionResNetV2 model on data from the SzUZGN set, it turned out after the initial experiments that transfer learning failed to bring the expected improvement in results compared to training the network from scratch. However, the acceleration of the learning process due to the pre-prepared model weights was a positive aspect of the transfer learning method. The results of the process are presented in Fig. A.1 and A.2

Data after PCA dimensionality reduction

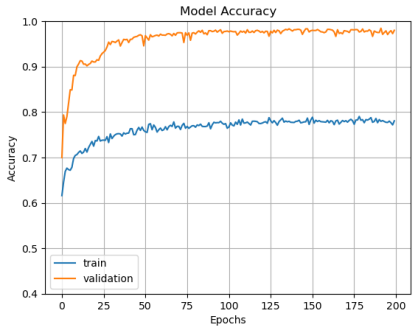
One of the most popular dimensionality reduction methods is principal component analysis (PCA). It is important to first examine the data in order to properly perform PCA analysis. The first issue concerns the number of observations, which in both databases exceeds 10,000, while the number of features is lower than 500. These two parameters made the JMP Pro (SAS Institute Inc. 1989-2022) application suggest the PCA method of pairwise comparisons. Another necessary parameter that needs to be determined is the number of main components. This number is chosen arbitrarily, but there are approaches, such as the Kaiser criterion (Kaiser 1960), which may help to make the right decision. It involves selecting only those main components whose eigenvalues exceed 1. This means that the examined component contributes at least the same information to the model as a single variable would. JMP Pro (SAS Institute Inc. 1989-2022) allows PCA analysis using a correlation matrix, which makes it unnecessary to normalize the input data.



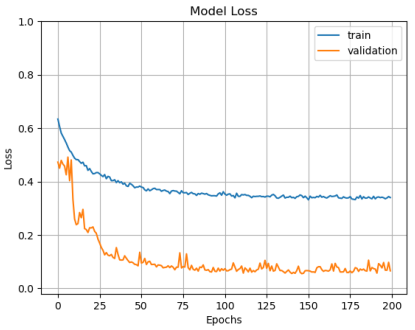
(a) Model accuracy in subsequent epochs



(b) Model loss in subsequent epochs



(c) Model accuracy in subsequent epochs

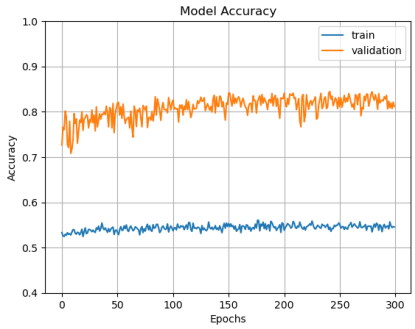


(d) Model loss in subsequent epochs

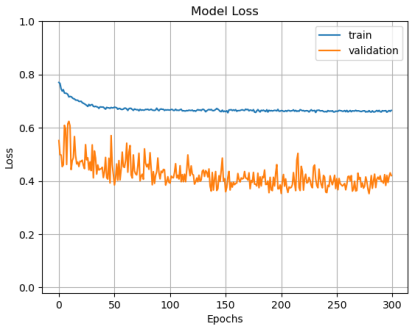
Figure A.1: Training process of the InceptionResNetV2 model on data from the BreakHis + GZG set

Principal components analysis on the BreakHis + GZG set The first part of the analysis concerns the BreakHis + GZG set. Of the 251 selected manual features, the PCA analysis revealed 29 significant principal components taking into account the Kaiser criterion. The selected principal components explain 89.413% of the variance of all original features. The next stage of the analysis was to perform PCA for a set of deep features. The use of the Kaiser criterion allowed for obtaining 3 main components that explain 89.615% of the variance of the remaining variables. Therefore, based on 25 deep features, the PCA method determined only 3 main components. Tab. A.1 shows the results of classification based on principal components for the test data.

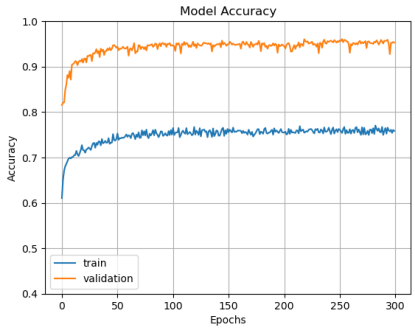
Based on the results obtained, it can be seen that when compared to models built on sets before PCA reduction, almost all classification results were slightly improved after the use of the PCA method for dimensionality reduction. The only exception was the k-NN classifier, which achieved better results for sets before PCA reduction. Both the improvement and the deterioration of the results after using the PCA method were very small, so the main advantage of dimensionality reduction by means of the PCA method is a much lower time needed to obtain



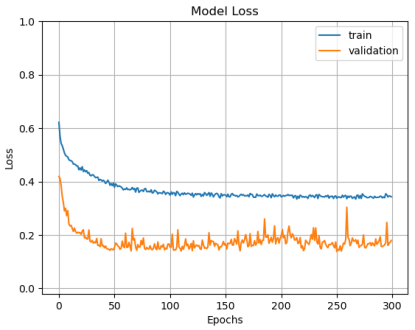
(a) Model accuracy in subsequent epochs



(b) Model loss in subsequent epochs



(c) Model accuracy in subsequent epochs



(d) Model loss in subsequent epochs

Figure A.2: Training process of the InceptionResNetV2 model on data from the SzUZG set

Table A.1: Classification results for test data for models trained on the BreakHis+GZG set after reducing the dimensionality of the feature set using the PCA method

Classifier	ACC [Manual]	ACC [Deep]	ACC [Fusion]
RL	0.7303	0.7926	0.7958
DT	0.7228	0.7871	0.7940
RF	0.7642	0.7857	0.7981
BT	0.7569	0.7918	0.7971
NB	0.7258	0.7950	0.7729
k-NN	0.7233	0.7958	0.7739
SVM	0.7571	0.7917	0.7903
NC	0.7342	0.7951	0.7937

the results, while the disadvantage is the deterioration of the interpretation of the results due to the replacement of the original features with the main components.

Analysis of principal components on the SzUZG set In order to enable comparison of the results with the data from the previous sections, the experiment was per-

formed separately for manual features and deep features. As a result of processing manual features, the number of main components selected based on the Kaiser criterion amounted to 29. This number explains 89.95% of the variance of all features. Therefore, the PCA method led to the reduction of the space from 251 features to 29. In the case of deep features, the number of main components that meet the criterion is 2. The selected main components explain 90.48% of the variance of all features. In the case of deep features, the dimensionality was reduced from 25 to 2 dimensions. Dimensionality reduction will most likely not improve classification results on training data, but it may contribute to improving results on test data by reducing the "curse of dimensionality". Therefore, the classification results for test data for the classifiers used in subsection are presented here in Fig. 4.8.3.3.

Table A.2: Classification results for test data for models trained on the SzUZG set after reducing the dimensionality of the feature set using the PCA method

Classifier	ACC [Manual]	ACC [Deep]	ACC [Fusion]
RL	0.6590	0.7271	0.7166
DT	0.6469	0.7191	0.7234
RF	0.6596	0.7202	0.7187
BT	0.6612	0.7209	0.7142
NB	0.6563	0.7271	0.7210
k-NN	0.6566	0.7209	0.6996
SVM	0.6563	0.7224	0.7057
NC	0.6580	0.7269	0.7192

Reducing the dimensionality of PCA on the SzUZG set contributed to a slight overall improvement of models based on decision trees (Tab. A.2). Also, improvement in classification results on a set of deep features is noteworthy. The research shows that two main components are sufficient to replace 25 features obtained from 5 different neural networks. It can therefore be assumed that all neural networks have similar characteristics of the obtained deep features. This conclusion leads to the need to perform an additional experiment involving the use of one neural network to obtain a larger number of features and possible reduction to the essential features. The success of such an experiment would also reduce the complexity of the model as well as the time needed to build it. Among the negative effects of the PCA analysis, it should be mentioned that it significantly worsened the maximum results obtained for the classifier models and, therefore, failed to contribute to obtaining better solutions.

Taking into account the best average results, it deteriorated for the set of manual features, as in the case of classifiers built on feature fusion. The most interesting thing, however, is that the highest average classification result for the set of deep features remained at the same level as before the PCA analysis and was achieved for the same classifier.

Data after performing exploratory feature analysis

Before performing the classification, the existing set of variables was explored in this study, which consisted of verifying the existence of linear relationships between variables. Detection of the value of the VIF variance inflation factor at a level exceeding 10 resulted in the exclusion of such a variable from the set of variables. The next step of the analysis was to examine the variables for skewness of the distributions and, if such skewness was detected, to attempt to reduce it using variable transformation methods. The last stage consisted of detecting and eliminating all outlier data from the set.

Data mining on the BreakHis + GZG set The performed analysis led to a reduction in the dimensionality of the set of manual features from 255 to 94. The set of deep features decreased from 25 to 7. The total number of features dropped from 276 to 101.

Table A.3: Classification results for test data for models trained on the BreakHis + GZG set after reducing the dimensionality of the feature set using VIF, distribution and outlier analysis

Classifier	ACC [Manual]	ACC [Deep]	ACC [Fusion]
RL	0.6942	0.7833	0.7775
DT	0.6910	0.7669	0.7728
RF	0.7194	0.7750	0.7842
DW	0.7165	0.7799	0.7699
NB	0.7409	0.7962	0.7897
k-NN	0.7046	0.7924	0.7715
SVM	0.7387	0.7886	0.7765
NC	0.7191	0.7831	0.7665

Data mining on the SzUZG set As a result of the analysis, out of 251 deep features, 97 remained in the database, with most of the remaining features subject to logarithmic transformation due to the asymmetry of distributions. Of the deep features, only 9 out of 25 remained in the database. After the analysis, the total number of features decreased from 276 to 106. The performed experiments failed to result in an overall improvement of the models. The NC model is particularly noteworthy (Tab. A.3 and Tab. A.4), because thanks to the use of exploratory analysis, the obtained classification accuracy results for both sets of test data amounted to above 0.7600. Unfortunately, this is still a result that is almost 3% worse than the models built on data before the analysis.

Table A.4: Classification results for test data for models trained on the SzUZG set after reducing the dimensionality of the feature set using VIF analysis, distributions and outliers

Classifier	ACC [Manual]	ACC [Deep]	ACC [Fusion]
RL	0.7199	0.7112	0.7278
DT	0.6468	0.7094	0.6610
RF	0.6823	0.7071	0.7148
DW	0.6613	0.7115	0.7142
NB	0.6823	0.7190	0.7214
k-NN	0.6444	0.7059	0.6920
SVM	0.7000	0.7146	0.7363
NC	0.7335	0.7118	0.7634

Appendix B

HARDWARE AND SOFTWARE CONFIGURATION

The following hardware and software configuration was used to perform the tests:

- Graphics card: Nvidia GeForce RTX 3060 12GB VRAM
- Processor: Intel i5-10600K, 6 cores, 12 threads.
- Motherboard: ASRock Z490M-ITX/ac
- RAM: 32 GB
- Disk: SSD 500 GB
- System: Ubuntu Linux, Windows 10

To manually select cell nuclei:

- Wacom Cintiq Pro 24 graphics tablet

Software:

- Python language including: Anaconda, Keras, Tensorflow, ScikitLearn, Numpy, Matplotlib, OpenCV, Pillow
- JMP Pro (SAS Institute Inc. 1989-2022)

Appendix C

U-NET STRUCTURE FOR SEGMENTATION

```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                                Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)                        [(None, 464, 704, 3  0)]         []
 conv2d (Conv2D)                             (None, 232, 352, 32  896)        ['input_1[0][0]']
 batch_normalization (BatchNormalization)    (None, 232, 352, 32  128)        ['conv2d[0][0]']
 activation (Activation)                     (None, 232, 352, 32  0)          ['batch_normalization[0][0]']
 activation_1 (Activation)                   (None, 232, 352, 32  0)          ['activation[0][0]']
 separable_conv2d (SeparableConv2D)          (None, 232, 352, 64  2400)       ['activation_1[0][0]']
 activation_2 (Activation)                   (None, 232, 352, 64  0)          ['batch_normalization_1[0][0]']
 separable_conv2d_1 (SeparableConv2D)        (None, 232, 352, 64  4736)       ['activation_2[0][0]']
 batch_normalization_2 (BatchNormalization)  (None, 232, 352, 64  256)        ['separable_conv2d_1[0][0]']
 max_pooling2d (MaxPooling2D)                (None, 116, 176, 64  0)          ['batch_normalization_2[0][0]']
 conv2d_1 (Conv2D)                           (None, 116, 176, 64  2112)       ['activation[0][0]']
 add (Add)                                   (None, 116, 176, 64  0)          ['max_pooling2d[0][0]','conv2d_1[0][0]']
 activation_3 (Activation)                   (None, 116, 176, 64  0)          ['add[0][0]']
 separable_conv2d_2 (SeparableConv2D)        (None, 116, 176, 12  88968)      ['activation_3[0][0]']
 batch_normalization_3 (BatchNormalization)  (None, 116, 176, 12  5128)       ['separable_conv2d_2[0][0]']
 activation_4 (Activation)                   (None, 116, 176, 12  08)         ['batch_normalization_3[0][0]']
 separable_conv2d_3 (SeparableConv2D)        (None, 116, 176, 12  176648)     ['activation_4[0][0]']
 batch_normalization_4 (BatchNormalization)  (None, 116, 176, 12  5128)       ['separable_conv2d_3[0][0]']
 max_pooling2d_1 (MaxPooling2D)              (None, 58, 88, 128)  0           ['batch_normalization_4[0][0]']
 conv2d_2 (Conv2D)                           (None, 58, 88, 128)  8320        ['add[0][0]']
 add_1 (Add)                                 (None, 58, 88, 128)  0           ['max_pooling2d_1[0][0]','conv2d_2[0][0]']
 activation_5 (Activation)                   (None, 58, 88, 128)  0           ['add_1[0][0]']
 separable_conv2d_4 (SeparableConv2D)        (None, 58, 88, 256)  34176       ['activation_5[0][0]']
 batch_normalization_5 (BatchNormalization)  (None, 58, 88, 256)  1024        ['separable_conv2d_4[0][0]']
 activation_6 (Activation)                   (None, 58, 88, 256)  0           ['batch_normalization_5[0][0]']
 separable_conv2d_5 (SeparableConv2d)        (None, 58, 88, 256)  68096       ['activation_6[0][0]']
 batch_normalization_6 (BatchNormalization)  (None, 58, 88, 256)  1024        ['separable_conv2d_5[0][0]']
 max_pooling2d_2 (MaxPooling2D)              (None, 29, 44, 256)  0           ['batch_normalization_6[0][0]']
 conv2d_3 (Conv2D)                           (None, 29, 44, 256)  33024       ['add_1[0][0]']
 add_2 (Add)                                 (None, 29, 44, 256)  0           ['max_pooling2d_2[0][0]','conv2d_3[0][0]']
 activation_7 (Activation)                   (None, 29, 44, 256)  0           ['add_2[0][0]']
 conv2d_transposeose) (Conv2DTransp)         (None, 29, 44, 256)  590080      ['activation_7[0][0]']
 batch_normalization_7 (BatchNormalization)  (None, 29, 44, 256)  1024        ['conv2d_transpose[0][0]']
 activation_8 (Activation)                   (None, 29, 44, 256)  0           ['batch_normalization_7[0][0]']
 conv2d_transpose_1 (Conv2DTranspose)        (None, 29, 44, 256)  590080      ['activation_8[0][0]']
 batch_normalization_8 (BatchNormalization)  (None, 29, 44, 256)  1024        ['conv2d_transpose_1[0][0]']
 up_sampling2d_1 (UpSampling2D)              (None, 58, 88, 256)  0           ['add_2[0][0]']
 up_sampling2d (UpSampling2D)                (None, 58, 88, 256)  0           ['batch_normalization_8[0][0]']
 conv2d_4 (Conv2D)                           (None, 58, 88, 256)  65792       ['up_sampling2d_1[0][0]']
 add_3 (Add)                                 (None, 58, 88, 256)  0           ['up_sampling2d[0][0]','conv2d_4[0][0]']
 activation_9 (Activation)                   (None, 58, 88, 256)  0           ['add_3[0][0]']
 conv2d_transpose_2 (Conv2DTranspose)        (None, 58, 88, 128)  295040      ['activation_9[0][0]']
 batch_normalization_9 (BatchNormalization)  (None, 58, 88, 128)  512         ['conv2d_transpose_2[0][0]']
 activation_10 (Activation)                  (None, 58, 88, 128)  0           ['batch_normalization_9[0][0]']
 conv2d_transpose_3 (Conv2DTranspose)        (None, 58, 88, 128)  147584      ['activation_10[0][0]']
 batch_normalization_10 (BatchNormalization) (None, 58, 88, 128)  512         ['conv2d_transpose_3[0][0]']
 up_sampling2d_3 (UpSampling2D)              (None, 116, 176, 25  06)         ['add_3[0][0]']
 up_sampling2d_2 (UpSampling2D)              (None, 116, 176, 12  08)         ['batch_normalization_10[0][0]']
 conv2d_5 (Conv2D)                           (None, 116, 176, 12  328968)     ['up_sampling2d_3[0][0]']
 add_4 (Add)                                 (None, 116, 176, 12  08          ['up_sampling2d_2[0][0]','conv2d_5[0][0]']
 activation_11 (Activation)                  (None, 116, 176, 12  08)         ['add_4[0][0]']
 conv2d_transpose_4 (Conv2DTranspose)        (None, 116, 176, 64  73792)      ['activation_11[0][0]']
 batch_normalization_11 (BatchNormalization) (None, 116, 176, 64  256)        ['conv2d_transpose_4[0][0]']
 activation_12 (Activation)                  (None, 116, 176, 64  0)          ['batch_normalization_11[0][0]']
 conv2d_transpose_5 (Conv2DTranspose)        (None, 116, 176, 64  36928)      ['activation_12[0][0]']
 batch_normalization_12 (BatchNormalization) (None, 116, 176, 64  256)        ['conv2d_transpose_5[0][0]']
 up_sampling2d_5 (UpSampling2D)              (None, 232, 352, 12  08)         ['add_4[0][0]']
 up_sampling2d_4 (UpSampling2D)              (None, 232, 352, 64  0)          ['batch_normalization_12[0][0]']
 conv2d_6 (Conv2D)                           (None, 232, 352, 64  8256)       ['up_sampling2d_5[0][0]']
 add_5 (Add)                                 (None, 232, 352, 64  0)          ['up_sampling2d_4[0][0]','conv2d_6[0][0]']
 activation_13 (Activation)                  (None, 232, 352, 64  0)          ['add_5[0][0]']
 conv2d_transpose_6 (Conv2DTranspose)        (None, 232, 352, 32  18464)      ['activation_13[0][0]']
 batch_normalization_13 (BatchNormalization) (None, 232, 352, 32  128)        ['conv2d_transpose_6[0][0]']
 activation_14 (Activation)                  (None, 232, 352, 32  0)          ['batch_normalization_13[0][0]']
 conv2d_transpose_7 (Conv2DTranspose)        (None, 232, 352, 32  9248)       ['activation_14[0][0]']
 batch_normalization_14 (BatchNormalization) (None, 232, 352, 32  128)        ['conv2d_transpose_7[0][0]']
```

```
 up_sampling2d_7 (UpSampling2D)              (None, 464, 704, 64  0)          ['add_5[0][0]']
 up_sampling2d_6 (UpSampling2D)              (None, 464, 704, 32  0)          ['batch_normalization_14[0][0]']
 conv2d_7 (Conv2D)                           (None, 464, 704, 32  2080)       ['up_sampling2d_7[0][0]']
 add_6 (Add)                                 (None, 464, 704, 32  0)          ['up_sampling2d_6[0][0]','conv2d_7[0][0]']
 conv2d_8 (Conv2D)                           (None, 464, 704, 3)  867         ['add_6[0][0]']
==================================================================================================
Total params: 2,058,979     Trainable params: 2,055,203     Non-trainable params: 3,776
```

The program code was created based on materials from the website `https://keras.io/`

Bibliography

Abadi, M. *et al.* (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org

Akaike, H. (1998). *Information Theory and an Extension of the Maximum Likelihood Principle*, in E. Parzen, K. Tanabe & G. Kitagawa (eds), Selected Papers of Hirotugu Akaike, Springer, New York, pp. 199–213.

Albon, C. (2019). *Uczenie Maszynowe w Pythonie*, Helion, Gliwice.

Alom, M. Z., Yakopcic, C., Taha, T. M. & Asari, V. K. (2018). *Nuclei Segmentation with Recurrent Residual Convolutional Neural Networks based U-Net (R2U-Net)*, IEEE National Aerospace and Electronics Conference (NAECON), pp. 228–233.

Anishiya, P. & Sasikala, M. (2016). *Segmentation and Localization of Epithelial Cells in the Histopathological Images of Stomach Adenocarcinoma*, International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET 2016), pp. 486–489.

Arooj, S. *et al.* (2022). *Breast Cancer Detection and Classification Empowered with Transfer Learning*, Frontiers in Public Health, Vol. 10.

Bulten, W. & Litjens, G. (2018). *Unsupervised Prostate Cancer Detection on H&E using Convolutional Adversarial Autoencoders*, Medical Imaging with Deep Learning.

Castelvecchi, D. (2016). *Can we open the black box of AI?*, Nature, Vol. 538, pp. 20–23.

Chennamsetty, S. S., Safwan, M. & Alex, V. (2018). *Classification of Breast Cancer Histology Image using Ensemble of Pre-trained Neural Networks*, in A. Campilho, F. Karray & B. ter Haar Romeny (eds), Image Analysis and Recognition, Springer International Publishing, Cham, pp. 804–811.

Chollet, F. (2017a). *Deep Learning with Python*, Manning Publications Co., New York.

Chollet, F. (2017b). *Xception: Deep Learning with Depthwise Separable Convolutions*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, Los Alamitos, pp. 1800–1807.

Cui, Y. & Hu, J. (2016). *Self-adjusting Nuclei Segmentation (SANS) of Hematoxylin-Eosin Stained Histopathological Breast Cancer Images*, IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2016), pp. 956–963.

Cui, Y. *et al.* (2019). *A Deep Learning Algorithm for One-step Contour Aware Nuclei Segmentation of Histopathology Images*, Medical & Biological Engineering & Computing, Vol. 57, No. 9, pp. 2027–2043.

Dee, F. R. *et al.* (2007). *Utility of 2-D and 3-D Virtual Microscopy in Cervical Cytology Education and Testing*, Acta Cytologica, Vol. 51, No. 4, pp. 523–529.

Deng, J. *et al.* (2009). *ImageNet: A large-scale hierarchical image database*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 248–255.

Dice, L. R. (1945). *Measures of the Amount of Ecologic Association Between Species*, Ecology, Vol. 26, No. 3, pp. 297–302.

Donnelly, A. D. *et al.* (2013). *Optimal Z-axis Scanning Parameters for Gynecologic Cytology Specimens*, Journal of Pathology Informatics, Vol. 4, No. 1, pp. 38.

Evans, T. *et al.* (2022). *The explainability paradox: Challenges for xAI in digital pathology*, Future Generation Computer Systems, Vol. 133, pp. 281–296.

Evered, A. & Dudding, N. (2011). *Accuracy and Perceptions of Virtual Microscopy Compared with Glass Slide Microscopy in Cervical cytology*, Cytopathology, Vol. 22, No. 2, pp. 82–87.

Fatakdawala, H. *et al.* (2010). *Expectation–Maximization-Driven Geodesic Active Contour With Overlap Resolution (EMaGACOR): Application to Lymphocyte Segmentation on Breast Cancer Histopathology*, IEEE Transactions on Biomedical Engineering, Vol. 57, No. 7, pp. 1676–1689.

Fondón, I. *et al.* (2018). *Automatic Classification of Tissue Malignancy for Breast Carcinoma Diagnosis*, Computers in Biology and Medicine, Vol. 96, pp. 41–51.

Fukuma, K. *et al.* (2016). *A Study on Feature Extraction and Disease Stage Classification for Glioma Pathology Images*, IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE Press, pp. 2150–2156.

Fukushima, K. (1980). *Neocognitron: A Self-organizing Neural Network Model for A Mechanism of Pattern Recognition Unaffected By Shift in Position*, Biological Cybernetics, Vol. 36, No. 4, pp. 193–202.

Gagnon, M. *et al.* (2004). *Comparison of Cytology Proficiency Testing*, Acta Cytologica, Vol. 48, No. 6, pp. 788–794.

Galloway, M. M. (1975). *Texture Analysis Using Gray Level Run Lengths*, Computer Graphics and Image Processing, Vol. 4, No. 2, pp. 172–179.

Géron, A. (2020). *Uczenie Maszynowe z Użyciem Scikit-Learn i TensorFlow*, Helion, Gliwice.

Gonzalez, R. J. & Woods, R. E. (2017). *Digital Image Processing, 4th edition*, Pearson, New York.

Hall, D. *et al.* (2015). *Evaluation of Features for Leaf Classification in Challenging Conditions*, IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 797–804.

Hanna, M. G. *et al.* (2017). *Comparison of glass slides and various digital-slide modalities for cytopathology screening and interpretation*, Cancer Cytopathology, Vol. 125, No. 9, pp. 701–709.

Haralick, R., Shanmugam, K. & Dinstein, I. (1973). *Textural Features for Image Classification*, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 3, No. 6, pp. 610–621.

Hayakawa, T. *et al.* (2021). *Computational Nuclei Segmentation Methods in Digital Pathology: A Survey*, Archives of Computational Methods in Engineering, Vol. 28, No. 1, pp. 1–13.

He, K., Gkioxari, G., Dollár, P. & Girshick, R. (2017). *Mask R-CNN*, IEEE International Conference on Computer Vision (ICCV), pp. 2980–2988.

He, K., Zhang, X., Ren, S. & Sun, J. (2016a). *Deep Residual Learning for Image Recognition*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778.

He, K., Zhang, X., Ren, S. & Sun, J. (2016b). *Identity Mappings in Deep Residual Networks*, in B. Leibe, J. Matas, N. Sebe & M. Welling (eds), Computer Vision – ECCV 2016, Springer International Publishing, Cham, pp. 630–645.

Hoerl, A. E. (1962). *Application of Ridge Analysis to Regression Problems*, Chemical Engineering Progress, Vol. 58, No. 3, pp. 54–59.

Hou, L. *et al.* (2016). *Automatic Histopathology Image Analysis with CNNs*, New York Scientific Data Summit (NYSDS), , pp. 1–6.

Huang, G., Liu, Z., Maaten, L. V. D. & Weinberger, K. Q. (2017). *Densely Connected Convolutional Networks*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, Los Alamitos, pp. 2261–2269.

Ilse, M., Tomczak, J. & Welling, M. (2018). *Attention-based Deep Multiple Instance Learning*, in J. Dy & A. Krause (eds), Proceedings of the 35th International Conference on Machine Learning, PMLR, Vol. 80 of Proceedings of Machine Learning Research, pp. 2127–2136.

Jaccard, P. (1912). *The Distribution of the Flora in the Alpine Zone.1*, New Phytologist, Vol. 11, No. 2, pp. 37–50.

Jantzen, J., Norup, J., Dounias, G. & Bjerregaard, B. (2005). *Pap-smear Benchmark Data For Pattern Classification*, Nature inspired Smart Information Systems (NiSIS), pp. 1–9.

Jassem, J. & Krzakowski, M. (2018). *Breast Cancer*, Oncology in Clinical Practice, Vol. 14, No. 4, pp. 171–215.

Jeleń, Ł., Fevens, T. & Krzyżak, A. (2008). *Classification of Breast Cancer Malignancy Using Cytological Images of Fine Needle Aspiration Biopsies*, International Journal of Applied Mathematics and Computer Science, Vol. 18, No. 1, pp. 75–83.

Kaiser, H. F. (1960). *The Application of Electronic Computers to Factor Analysis*, Educational and Psychological Measurement, Vol. 20, No. 1, pp. 141–151.

Kashif, M. N. *et al.* (2016). *Handcrafted Features with Convolutional Neural Networks for Detection of Tumor Cells in Histology Images*, IEEE 13th International Symposium on Biomedical Imaging (ISBI), pp. 1029–1032.

Kowal, M., Skobel, M., Gramacki, A. & Korbicz, J. (2021). *Breast Cancer Nuclei Segmentation and Classification Based on a Deep Learning Approach*, International Journal of Applied Mathematics and Computer Science, Vol. 31, No. 1, pp. 85–106.

Kowal, M., Skobel, M. & Nowicki, N. (2018). *The Feature Selection Problem in Computer–Assisted Cytology*, International Journal of Applied Mathematics and Computer Science, Vol. 28, pp. 759–770.

Kowal, M. *et al.* (2013). *Computer-aided Diagnosis of Breast Cancer Based on Fine Needle Biopsy Microscopic Images*, Computers in Biology and Medicine, Vol. 43, No. 10, pp. 1563–1572.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). *ImageNet Classification with Deep Convolutional Neural Networks*, Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, NIPS'12, Curran Associates Inc., Red Hook, pp. 1097–1105.

Kumar, V. *et al.* (2020). *Prediction of Malignant and Benign Breast Cancer: A Data Mining Approach in Healthcare Applications*, in S. Borah, V. Emilia Balas & Z. Polkowski (eds), Advances in Data Science and Management, Springer Singapore, Singapore, pp. 435–442.

Kwok, S. (2018). *Multiclass Classification of Breast Cancer in Whole-Slide Images*, in A. Campilho, F. Karray & B. ter Haar Romeny (eds), Image Analysis and Recognition, Springer International Publishing, Cham, pp. 931–940.

Lagree, A. *et al.* (2021). *A Review and Comparison of Breast Tumor Cell Nuclei Segmentation Performances Using Deep Convolutional Neural Networks*, Scientific Reports, Vol. 11, No. 1, pp. 8025.

LeCun, Y. & Bengio, Y. (1995). *Convolutional Networks for Images, Speech, and Time Series*, in M. A. Arbib (ed.), Handbook of Brain Theory and Neural Networks, MIT Press, pp. 33–61.

Lecun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998). *Gradient-based Learning Applied to Document Recognition*, Proceedings of the IEEE, Vol. 86, No. 11, pp. 2278–2324.

Linkert, M. *et al.* (2010). *Metadata Matters: Access to Image Data in the Real World*, Journal of Cell Biology, Vol. 189, No. 5, pp. 777–782.

Lu, C., Mahmood, M., Jha, N. & Mandal, M. (2012). *A Robust Automatic Nuclei Segmentation Technique for Quantitative Histopathological Image Analysis*, Analytical and Quantitative Cytology and Histology, Vol. 34, No. 6, pp. 296–308.

Maier-Hein, L. *et al.* (2018). *Why Rankings of Biomedical Image Analysis Competitions Should Be Interpreted with Care*, Nature Communications, Vol. 9, No. 1, pp. 5217.

Mangasarian, O. L., Street, W. N. & Wolberg, W. H. (1995). *Breast Cancer Diagnosis and Prognosis via Linear Programming*, Operations Research, Vol. 43, No. 4, pp. 570–577.

Minsky, M. & Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry*, MIT Press, Cambridge.

Naji, M. A. *et al.* (2021). *Machine Learning Algorithms For Breast Cancer Prediction And Diagnosis*, Procedia Computer Science, Vol. 191, pp. 487–492.

Naylor, P., Laé, M., Reyal, F. & Walter, T. (2019). *Segmentation of Nuclei in Histopathology Images by Deep Regression of the Distance Map*, IEEE Transactions on Medical Imaging, Vol. 38, No. 2, pp. 448–459.

Neghina, M. *et al.* (2016). *Automatic Detection of Cervical Cells in Pap-smear Images Using Polar Transform and K-Means Segmentation*, Sixth International

Conference on Image Processing Theory, Tools and Applications (IPTA), pp. 1–6.

Niazi, M. K. K. *et al.* (2017). *Visually Meaningful Histopathological Features for Automatic Grading of Prostate Cancer*, IEEE Journal of Biomedical and Health Informatics, Vol. 21, No. 4, pp. 1027–1038.

Otsu, N. (1979). *A Threshold Selection Method from Gray-Level Histograms*, IEEE Transitions Systems, Man, and Cybernetics, Vol. 9, No. 1, pp. 62–66.

Paszke, A. *et al.* (2019). *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, Advances in Neural Information Processing Systems 32, Curran Associates, Inc., pp. 8024–8035.

Pearson, K. (1895). *Note on Regression and Inheritance in the Case of Two Parents*, Proceedings of the Royal Society of London Series I, Vol. 58, pp. 240–242.

Petushi, S. *et al.* (2006). *Large-scale Computations on Histology Images Reveal Grade-differentiating Parameters for Breast Cancer*, BMC Medical Imaging, Vol. 6, No. 1, pp. 14.

Phoulady, H. A., Goldgof, D., Hall, L. & Mouton, P. (2016). *A New Approach to Detect and Segment Overlapping Cells in Multi-layer Cervical Cell Volume Images*, IEEE 13th International Symposium on Biomedical Imaging (ISBI), pp. 201–204.

Phoulady, H. A. *et al.* (2016). *Automatic Quantification and Classification of Cervical Cancer via Adaptive Nucleus Shape Modeling*, IEEE International Conference on Image Processing (ICIP), pp. 2658–2662.

Pilleron, S. *et al.* (2021). *Estimated Global Cancer Incidence in the Oldest Adults in 2018 and Projections to 2050*, International Journal of Cancer, Vol. 148, No. 3, pp. 601–608.

Qi, X., Xing, F., Foran, D. J. & Yang, L. (2011). *Robust Segmentation of Overlapping Cells in Histopathology Specimens Using Parallel Seed Detection and Repulsive Level Set*, IEEE Transactions on Biomedical Engineering, Vol. 59, No. 3, pp. 754–765.

Ragothaman, S., Narasimhan, S., Basavaraj, M. G. & Dewar, R. (2016). *Unsupervised Segmentation of Cervical Cell Images Using Gaussian Mixture Model*, IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1374–1379.

Rajaganesan, S. *et al.* (2021). *Comparative Assessment of Digital Pathology Systems for Primary Diagnosis*, Journal of Pathology Informatics, Vol. 12, No. 1, pp. 25.

Rajyalakshmi, U., Rao, S. K. & Prasad, K. S. (2017). *Supervised Classification of Breast Cancer Malignancy Using Integrated Modified Marker Controlled Watershed Approach*, IEEE 7th International Advance Computing Conference (IACC 2017), pp. 584–589.

Raschka, S. & Mirjalili, V. (2019). *Python. Uczenie Maszynowe. Wydanie II*, Helion, Gliwice.

Ronneberger, O., Fischer, P. & Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*, in N. Navab, J. Hornegger, W. M. Wells & A. F. Frangi (eds), Medical Image Computing and Computer-Assisted In-

tervention (MICCAI 2015, Springer International Publishing, Cham, pp. 234–241.

Ruifrok, A. & Johnston, D. (2001). *Quantification of Histochemical Staining by Color Deconvolution*, Analytical and Quantitative Cytology and Histology, Vol. 23, No. 4, pp. 291–299.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986). *Learning Representations by Back-propagating Errors*, Nature, Vol. 323, No. 6088, pp. 533–536.

Saha, R., Bajger, M. & Lee, G. (2016). *Spatial Shape Constrained Fuzzy C-Means (FCM) Clustering for Nucleus Segmentation in Pap Smear Images*, International Conference on Digital Image Computing: Techniques and Applications (DICTA), pp. 1–8.

Sakib, S. *et al.* (2022). *Breast Cancer Detection and Classification: A Comparative Analysis Using Machine Learning Algorithms*, in V. Bindhu, J. M. R. S. Tavares & K.-L. Du (eds), Proceedings of Third International Conference on Communication, Computing and Electronics Systems, Springer Singapore, Singapur, pp. 703–717.

Salvi, M. & Molinari, F. (2018). *Multi-tissue and Multi-scale Approach for Nuclei Segmentation in H&E Stained Images*, BioMedical Engineering OnLine, Vol. 17.

SAS Institute Inc. (1989-2022). *JMP®, Version 16.0.0.*

Schindelin, J. *et al.* (2012). *Fiji: an Open-source Platform for Biological-image Analysis*, Nature Methods, Vol. 9, No. 7, pp. 676–682.

Schwarz, G. (1978). *Estimating the Dimension of a Model*, The Annals of Statistics, Vol. 6, No. 2, pp. 461–464.

Shi, P., Zhong, J., Huang, R. & Lin, J.-J. (2016). *Automated Quantitative Image Analysis of Hematoxylin-Eosin Staining Slides in Lymphoma Based on Hierarchical Kmeans Clustering*, 8th International Conference on Information Technology in Medicine and Education (ITME 2016), pp. 99–104.

Shu, J. *et al.* (2013). *Segmenting Overlapping Cell nuclei In Digital Histopathology Images*, 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 5445–5448.

Simonyan, K. & Zisserman, A. (2015). *Very Deep Vonvolutional Networks for Large-scale Image Recognition*, 3rd International Conference on Learning Representations (ICLR 2015), Computational and Biological Learning Society, pp. 1–14.

Sirinukunwattana, K. *et al.* (2016). *Locality Sensitive Deep Learning for Detection and Classification of Nuclei in Routine Colon Cancer Histology Images*, IEEE Transactions on Medical Imaging, Vol. 35, No. 5, pp. 1196–1206.

Skobel, M., Kowal, M. & Korbicz, J. (2020). *Breast Cancer Computer-Aided Diagnosis System Using k-NN Algorithm Based on Hausdorff Distance*, in J. Korbicz, R. Maniewski, K. Patan & M. Kowal (eds), Current Trends in Biomedical Engineering and Bioimages Analysis, Springer International Publishing, Cham, pp. 179–188.

Solanki, Y. S. *et al.* (2021). *A Hybrid Supervised Machine Learning Classifier System for Breast Cancer Prognosis Using Feature Selection and Data Imbalance Handling Approaches*, Electronics, Vol. 10, No. 6, pp. 699.

Sørensen, T. (1948). *A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vegetation on Danish Commons*, Biologiske skrifter, I kommission hos E. Munksgaard.

Sornapudi, S. *et al.* (2019). *Comparing Deep Learning Models for Multi-cell Classification in Liquid-based Cervical Cytology Image*, AMIA Annual Symposium Proceedings, Vol. 2019, pp. 820–827.

Spanhol, F. A., Oliveira, L. S., Petitjean, C. & Heutte, L. (2016). *A Dataset for Breast Cancer Histopathological Image Classification*, IEEE Transactions on Biomedical Engineering, Vol. 63, No. 7, pp. 1455–1462.

Spanhol, F. A. *et al.* (2017). *Deep Features for Breast Cancer Histopathological Image Classification*, IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 1868–1873.

Street, W. N., Wolberg, W. H. & Mangasarian, O. L. (1993). *Nuclear Feature Extraction for Breast Tumor Diagnosis*, in R. S. Acharya & D. B. Goldgof (eds), Biomedical Image Processing and Biomedical Visualization, International Society for Optics and PhotonicsSPIE, Vol. 1905, pp. 861–870.

Sung, H. *et al.* (2021). *Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries*, CA: A Cancer Journal for Clinicians, Vol. 71, No. 3, pp. 209–249.

Szegedy, C., Ioffe, S., Vanhoucke, V. & Alemi, A. A. (2017). *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*, Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI'17, AAAI Press, pp. 4278–4284.

Szegedy, C. *et al.* (2015). *Going Deeper with Convolutions*, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, Los Alamitos, pp. 1–9.

Szegedy, C. *et al.* (2016). *Rethinking the Inception Architecture for Computer Vision*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), pp. 2818–2826.

Tang, X. (1998). *Texture Information in Run-Length Matrices*, IEEE Transactions on Image Processing, Vol. 7, No. 11, pp. 1602–1609.

Tareef, A. *et al.* (2016). *Automatic Nuclei and Cytoplasm Segmentation of Leukocytes with Color and Texture-based Image Enhancement*, IEEE 13th International Symposium on Biomedical Imaging (ISBI), pp. 935–938.

Theano Development Team (2016). *Theano: A Python Framework for Fast Computation of Mathematical Expressions*, arXiv e-prints, Vol. abs/1605.02688.

Tibshirani, R. (1996). *Regression Shrinkage and Selection via the Lasso*, Journal of the Royal Statistical Society. Series B (Methodological), Vol. 58, No. 1, pp. 267–288.

Tripathi, S. & Singh, S. K. (2020). *Ensembling Handcrafted Features with Deep Features: an Analytical Study for Classification of Routine Colon Cancer*

Histopathological Nuclei Images, Multimedia Tools and Applications, Vol. 79, No. 47, pp. 34931–34954.

Van Es, S. L. (2018). *Digital Pathology: Semper Ad Meliora*, Pathology, Vol. 51, No. 1, pp. 1–10.

Veta, M. *et al.* (2011). *Marker-controlled Watershed Segmentation of Nuclei in H&E Stained Breast Cancer Biopsy Images*, IEEE International Symposium on Biomedical Imaging: From Nano to Macro, pp. 618–621.

Veta, M. *et al.* (2013). *Automatic Nuclei Segmentation in H&E Stained Breast Cancer Histopathology Images*, PLOS ONE, Vol. 8, No. 7, pp. e70221.

Wang, H. *et al.* (2014). *Mitosis Detection in Breast Cancer Pathology Images by Combining Handcrafted and Convolutional Neural Network Features*, Journal of Medical Imaging, Vol. 1, No. 3, pp. 034003.

Wang, M. *et al.* (2017). *A Sensitivity and Specificity Comparison of Fine Needle Aspiration Cytology and Core Needle Biopsy in Evaluation of Suspicious Breast Lesions: A Systematic Review and Meta-analysis*, The Breast, Vol. 31, pp. 157–166.

Wang, X. *et al.* (2020). *Weakly Supervised Deep Learning for Whole Slide Lung Cancer Image Analysis*, IEEE Transactions on Cybernetics, Vol. 50, No. 9, pp. 3950–3962.

Win, K. Y. & Choomchuay, S. (2017). *Automated Segmentation of Cell Nuclei in Cytology Pleural Fluid Images Using OTSU Thresholding*, International Conference on Digital Arts, Media and Technology (ICDAMT), pp. 14–18.

Wolberg, W. H., Street, W. & Mangasarian, O. (1994). *Machine Learning Techniques to Diagnose Breast Cancer from Image-processed Nuclear Features of Fine Needle Aspirates*, Cancer Letters, Vol. 77, No. 2, pp. 163–171.

Yang, X., Li, H. & Zhou, X. (2006). *Nuclei Segmentation Using Marker-Controlled Watershed, Tracking Using Mean-Shift, and Kalman Filter in Time-Lapse Microscopy*, IEEE Transactions on Circuits and Systems I: Regular Papers, Vol. 53, No. 11, pp. 2405–2414.

Yedjou, C. G. *et al.* (2021). *Application of Machine Learning Algorithms in Breast Cancer Diagnosis and Classification*, International Journal of Science Academic Research, Vol. 2, No. 1, pp. 3081–3086.

Zarei, N. *et al.* (2017). *Automated Prostate Glandular and Nuclei Detection Using Hyperspectral Imaging*, IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017), pp. 1028–1031.

Zhang, L. *et al.* (2014). *Segmentation of Cytoplasm and Nuclei of Abnormal Cells in Cervical Cytology Using Global and Local Graph Cuts*, Computerized Medical Imaging and Graphics, Vol. 38, No. 5, pp. 369–380.

Zhang, L. *et al.* (2017). *DeepPap: Deep Convolutional Networks for Cervical Cell Classification*, IEEE Journal of Biomedical and Health Informatics, Vol. 21, No. 6, pp. 1633–1643.

Zhang, R., Du, L., Xiao, Q. & Liu, J. (2020). *Comparison of Backbones for Semantic Segmentation Network*, Journal of Physics: Conference Series, Vol. 1544, No. 1.

Zhang, Z., Wu, C., Coleman, S. & Kerr, D. (2020). *DENSE-INception U-net for Medical Image Segmentation*, Computer Methods and Programs in Biomedicine, Vol. 192.

Zhao, Z., Zhao, D., Yang, S. & Xu, L. (2023). *Image-Based Malware Classification Method with the AlexNet Convolutional Neural Network Model*, Security and Communication Networks, Vol. 2023.

Zou, H. & Hastie, T. (2005). *Regularization and Variable Selection via the Elastic Net*, Journal of the Royal Statistical Society. Series B (Statistical Methodology), Vol. 67, No. 2, pp. 301–320.

Index

aggregation (bagging), 61
AlexNet, 58

breast cancer, 12
BreakHis dataset, 26
boosting, 62

confusion matrix, 71
contamination (stacking), 62
convolutional neural networks, 57

deconvolution, 30
DenseNet, 60
detection, 32
Dice-Sørensen coefficient, 37
digital image, 24
dimensionality reduction, 63

embedded feature selection, 66
erosion, 39

feature extraction, 53
feature selection, 64
filter feature selection, 65

Gamma distribution, 68
GoogLeNet, 60

Hausdorff distance, 37
hyperbolic tangent, 57

instance segmentation, 32

Jaccard index, 36

machine learning, 56
marker-controlled watershed, 35
method of moments, 68

Otsu binarization, 30

random forest, 62
region of interest, 36
regularization, 66
ReLU, 57
ResNet, 60

semantic segmentation, 32
stochastic feature selection, 67

U-Net, 33
unipolar sigmoid, 57

VGG, 59
voting, 61

wrapper, 64

Xception, 61

List of figures

1.1 Diagram of the proposed approach 21
1.2 Deep feature acquisition scheme 22
1.3 Empirical distribution and fitted Gamma distribution 22
2.1 Description of the specimen at maximum magnification (benign case) 25
2.2 Maximum magnification of a portion of the virtual slide area . . . 27
2.3 Crop the image to 700 × 460 28
2.4 Scheme for preparing databases with images 29
2.5 The effect of image deconvolution using the H&E method using Fiji software 31
3.1 Differences between detection, semantic segmentation and instance segmentation 33
3.2 U-Net network diagram based on the article by Ronneberger . . . 33
3.3 Hybrid method of watershed segmentation using initial segmentation using a CNN network and detecting watershed markers using a second CNN network 35
3.4 Stages of transition from the real object to the ROI and binary mask 36
3.5 Graphical interpretation of DH 37
3.6 The course of training the U-Net model 39
3.7 Scheme of creating edges of cell nuclei of images in the training set 40
3.8 Training flow of the U-Net model for central points 41
3.9 Scheme of segmentation of cell nuclei 41
3.10 Example of segmentation results 42
3.11 Value distributions and statistics 43
3.12 Example image with a low segmentation evaluation score 44
3.13 An example image with an average segmentation evaluation result 45
3.14 Distributions of the number of features in groups 45
3.15 Boxplots of segmentation results to the center of origin of the samples 46
3.16 Matching segmentation results to the center of origin of the samples 49
3.17 Matching segmentation results to the created sets 50
3.18 Matching segmentation results to malignant and benign cases . . . 51
3.19 Matching the results of segmentation metrics to origin from sets (a, b) and cancer cases (c, d) 52
4.1 Feature extraction 53
4.2 Diagram of the field of artificial intelligence according to Chollet . 56
4.3 Example of CNN 58
4.4 AlexNet network 59
4.5 VGG16 network 59
4.6 Inception module 60
4.7 Deep feature generators 62

4.8 Feature selection - wrapped methods (wrappers) 64
4.9 Feature selection - filtering methods 65
4.10 Feature selection - embedded methods 66
4.11 Examples of empirical distributions 68
4.12 Schemat głównych eksperymentów 72
4.13 Scheme of experiments on CNN networks 73
4.14 Training process of the ResNet50 model for the BreakHis + GZG training set 75
4.15 Training process of the Resnet50 model for training data from the SzUZG set 75
4.16 Training process of the Resnet152V2 model for the BreakHis + GZG training set 76
4.17 Training process of the Resnet152V2 model for training data from the SzUZG set 77
4.18 Training process of the VGG16 model for training data from the BreakHis + GZG set 77
4.19 Training process of the VGG16 model for training data from the SzUZG set 78
4.20 Training process of the VGG19 model for the BreakHis + GZG training set 78
4.21 Training process of the VGG19 model for training data from the SzUZG set 79
4.22 Training process of the DenseNet121 model for training data from the BreakHis + GZG set 79
4.23 Training process of the DenseNet121 model for training data from the SzUZG set 80
4.24 Training process of the Xception model for the BreakHis + GZG training set 81
4.25 Training process of the Xception model for training data from the SzUZG set 81
4.26 Training process of the InceptionResNetV2 model for training data from the BreakHis + GZG set 82
4.27 Training process of the InceptionResNetV2 model for training data from the SzUZG set 83
4.28 Scheme of experiments on master classifiers 87
4.29 Sensitivity and precision graph for the best models trained on the BreakHis+ GZG set and tested on SzUZG 103
4.30 Sensitivity and precision graph for the best models trained on the SzUZG set and tested on BreakHis + GZG 103
4.31 Sensitivity and precision graph for the best models trained on the SzUZG set and tested on BreakHis 104
A.1 Training process of the InceptionResNetV2 model on data from the BreakHis + GZG set 111
A.2 Training process of the InceptionResNetV2 model on data from the SzUZG set 112

List of tables

4.1 List of extracted features of cell nuclei 55
4.2 Confusion matrix 71
4.3 Summary of image classification results in RGB space 84
4.4 Summary of image classification results after deconvolution 85
4.5 Summary of image classification results after segmentation 86
4.6 Regression with Elastic Net regularization and validation using the Bayes information criterion, trained on the BreakHis + GZG set and tested on the SzUZG set 88
4.7 Regression with Elastic Net regularization and validation using the Bayes information criterion, trained on the SzUZG set and tested on the BreakHis + GZG set 89
4.8 DT, trained on the BreakHis + GZG set, and tested on the SzUZG set 90
4.9 DT, trained on the SzUZG set, and tested on the BreakHis + GZG set 91
4.10 RF, trained on the BreakHis + GZG set, and tested on the SzUZG set 92
4.11 RF, trained on the SzUZG set, and tested on the BreakHis + GZG set 92
4.12 BT, trained on the BreakHis + GZG set, and tested on the SzUZG set 93
4.13 BT, trained on the SzUZG set, and tested on the BreakHis + GZG set 94
4.14 NB trained on the BreakHis + GZG set, and tested on the SzUZG set 95
4.15 NB, trained on the SzUZG set, and tested on the BreakHis + GZG set 95
4.16 k-NN classifier, trained on the BreakHis + GZG set, and tested on the SzUZG set 96
4.17 k-NN classifier, trained on the SzUZG set, and tested on the BreakHis + GZG set 97
4.18 SVM with a radial activation function, trained on the BreakHis + GZG set, and tested on the SzUZG set 98
4.19 SVM with a radial activation function, trained on the SzUZG set and tested on the BreakHis + GZG set 98
4.20 NC with a single neuron with TanH activation, trained on the zBreakHis + GZG set, and tested on the SzUZG set 99
4.21 NC with a single neuron with TanH activation, trained on the SzUZG set, and tested on the BreakHis + GZG set 100

4.22 Summary of the best classifiers based on accuracies obtained for test data divided into feature sets (Notation BreakHis + GZG shortened to BreakHis in the table) 102
A.1 Classification results for test data for models trained on the BreakHis+GZG set after reducing the dimensionality of the feature set using the PCA method 112
A.2 Classification results for test data for models trained on the SzUZG set after reducing the dimensionality of the feature set using the PCA method 113
A.3 Classification results for test data for models trained on the BreakHis + GZG set after reducing the dimensionality of the feature set using VIF, distribution and outlier analysis 114
A.4 Classification results for test data for models trained on the SzUZG set after reducing the dimensionality of the feature set using VIF analysis, distributions and outliers 115

Streszczenie

GŁĘBOKIE SIECI NEURONOWE W KLASYFIKACJI OBRAZÓW MEDYCZNYCH

Diagnostyka obrazowa jest jednym z najważniejszych zastosowań sztucznej inteligencji w obszarze medycyny. Mimo obserwowanych obecnie spektakularnych sukcesów sztucznych sieci neuronowych w dziedzinie przetwarzania języka naturalnego i analizy obrazów istnieją nadal trudne wyzwania, które muszą zostać rozwiązane, aby możliwe było wdrożenie sztucznej inteligencji w rutynową diagnostykę medyczną.

W ramach rozprawy podjęto problem klasyfikacji nowotworów piersi na podstawie obrazów histopatologicznych i cytologicznych. Literatura naukowa dostarcza nam bogatą bazę rozwiązań dla tego problemu, wskazując na głębokie sieci neuronowe jako aktualnie najlepsze rozwiązanie. Niestety, jeśli przyjrzymy się bliżej proponowanym modelom, to spostrzeżemy, że są one najczęściej uczone i testowane na zbiorze obrazów pochodzących z jednego ośrodka medycznego. Przeprowadzone testy wykazały, że wyuczone w ten sposób modele głębokich sieci neuronowych nie są w stanie nabyć odpowiednich zdolności uogólniających, aby mogły być stosowane do klasyfikacji obrazów podchodzących z innych ośrodków medycznych. Przyczyną takiego stanu jest zróżnicowanie pomiędzy obrazami pochodzącymi z różnych ośrodków medycznych. Mimo że procedury pozyskiwania obrazów są w pewien sposób standaryzowane, to jednak czynnik ludzki oraz techniczny powoduje, że w praktyce obrazy różnią się wieloma istotnymi cechami. Niestety, nie jest również możliwe zbudowanie odpowiednio bogatego i różnorodnego zbioru obrazów histopatologicznych lub cytologicznych nowotworu piersi, który zapewniłby uogólnianie wiedzy modeli na poziomie pozwalającym stosować je dla obrazów pochodzących z różnych ośrodków medycznych. Publicznie dostępne zbiory obrazów dla nowotworu piersi są niewielkie, zbyt jednorodne (reprezentują zwykle tylko kilkunastu lub kilkudziesięciu pacjentów) i jest ich stanowczo za mało.

Badania podjęte w rozprawie skupiły się na opracowaniu modelu pozwalającego klasyfikować nowotwory piersi na podstawie obrazów histopatologicznych lub cytologicznych pochodzących z jednego ośrodka medycznego, który jest skuteczny również dla danych pochodzących z innego ośrodka. Do realizacji tego projektu wykorzystano obrazy medyczne ze zbioru BreakHis z Brazylii oraz obrazy ze Szpitala Uniwersyteckiego w Zielonej Górze (SzUZG). Eksperymenty wykazały, że na pewnym poziomie normalizacji obrazów występuje transfer wiedzy umożliwiający zbudowanie uogólnionego skutecznego systemu klasyfikacji.

Wypracowane podejście składa się z czterech głównych rozwiązań. Przede wszystkim obrazy poddano standaryzacji z wykorzystaniem zaprojektowanej w tym celu hybrydowej metody segmentacji. Do realizacji tego kroku zaproponowano dwie sieci neuronowe typu U-Net oraz metodę wododziałową. Pierwsza z sieci neuronowych była odpowiedzialna ze segmentację semantyczną obrazów a druga za lokalizację środków poszczególnych jąder komórkowych. Metoda wododziałowa dokonywała fuzji informacji pozyskanych z sieci neuronowych, aby dokonać segmentacji poszczególnych instancji jąder komórkowych.

Kolejnym rozwiązaniem, które zaproponowano w ramach rozprawy, była fuzja cech manualnych z cechami głębokimi, aby zwiększyć odporność opisu obrazów na niejednorodność wewnątrzklasową obrazów. Miało to na celu nabycie przez budowany model zdolności uogólniających wykraczających poza obrazy pochodzące z jednego ośrodka medycznego. W tym celu konieczne było opracowanie zautomatyzowanego systemu do ekstrakcji cech manualnych w oparciu o segmentację jąder komórkowych. Wykorzystano do tego zadania hybrydową metodę segmentacji opracowaną na potrzeby standaryzacji obrazów. System do ekstrakcji cech głębokich powstał w oparciu o heterogeniczny zespół głębokich sieci neuronowych. Członkowie tego zespołu byli strojeni indywidualnie z wykorzystaniem tego samego zbioru obrazów. Ostatecznie z warstw pośrednich głębokich sieci neuronowych wyodrębniono bogate zestawy cech głębokich, które połączono w jeden zestaw wraz z cechami manualnymi.

W efekcie powstał bardzo liczny zestaw cech do opisu obrazów, który należało zredukować do cech istotnych przy klasyfikacji nowotworów piersi. Dlatego w ramach kolejnego kroku badawczego przetestowano wiele znanych metod selekcji cech. Niestety, ze względu na duży rozmiar wektora cech i stosunkowo dużą liczbę próbek standardowe metody redukcji wymiarowości okazały się niezmiernie kosztowne czasowo i obliczeniowo. Z tego powodu opracowano nowe rozwiązanie do selekcji cech. Opracowana metoda bazuje na przeszukiwaniu stochastycznym. Na wstępie poszukiwany jest rozkład optymalnej liczby cech na podstawie reprezentatywnego podzbioru obrazów. Pozwala to na etapie właściwej selekcji cech znacząco ograniczyć przestrzeń poszukiwań i w efekcie przyśpieszyć i polepszyć wyniki selekcji cech.

Ostatnim elementem opracowanego systemu jest nadrzędny klasyfikator, który na wejście otrzymuje zestaw cech ustalony przez opracowaną metodę selekcji cech. Rolę tego klasyfikatora pełni model regresji z regularyzacją L2. Model ten został wybrany w toku wielu eksperymentów porównujących efektywność różnych technik uczenia maszynowego.

Zważywszy że jednym z kluczowych zagadnień poruszonych w rozprawie była weryfikacja skuteczności działania zaproponowanego systemu zbudowanego na danych medycznych pochodzących z innego ośrodka badawczego niż dane testowe, przeprowadzone zostały kompleksowe badania. W wyniku przeprowadzonych eksperymentów wykazano, iż zaproponowana metoda uzyskuje średnią dokładność klasyfikacji obrazów na poziomie 79% co przekłada się na wynik średniej precyzji dla pojedynczego pacjenta na poziomie przekraczającym 90% dokładności.

Główne osiągnięcia rozprawy doktorskiej obejmują:

- Przygotowanie hybrydowej metody segmentacji obrazów cytologicznych i histopatologicznych na potrzeby standaryzacji obrazów i ekstrakcji cech manualnych jąder komórkowych.
- Przygotowanie systemu ekstrakcji cech do opisu obrazów z wykorzystaniem fuzji cech manualnych z cechami głębokimi uzyskanymi z heterogenicznego zespołu głębokich sieci neuronowych.
- Opracowanie metody stochastycznej selekcji cech z zestawu zawierającego cechy głębokie oraz wyekstrahowane manualnie.
- Przeprowadzanie kompleksowych badań weryfikujących efektywność opracowanych metod dla rzeczywistych obrazów pochodzących ze zbioru BreakHis oraz zbioru obrazów SzUZG.

Lecture Notes in Control and Computer Science

Editor-in-Chief: Józef KORBICZ

Vol. 28: Marcin Skobel
Deep Neural Networks in Medical Image Classification
138p. 2024 [978-83-7842-558-8]

Vol. 27: Marek Wróblewski
Quantum Computational Methods in Hybrid Classical-quantum Recommendation Systems
180p. 2023 [978-83-7842-530-4]

Vol. 26: Marcel Luzar
Dynamic Artificial Neural Networks in Designing Robust Fault Diagnosis Systems
154p. 2016 [978-83-7842-282-2]

Vol. 25: Andrzej Czajkowski
Fault Tolerant Control System Design Using Dynamic Neural Networks
140 p. 2016 [978-83-7842-263-1]

Vol. 24: Iwona Grobelna
Formal Verification of Logic Controller Specification by Means of Model Checking
168 p. 2013 [978-83-7842-060-6]

Vol. 23: Małgorzata Wiśniewska
Application of Hypergraphs in Decomposition of Discrete Systems
143 p. 2012 [978-83-7842-025-5]

Vol. 22: Mariusz Jacyno
Self-organising Agent Communities for Autonomic Computing
202 p. 2012 [978-83-7842-012-5]

Vol. 21: Błażej Cichy
Analysis and Control of Multi-dimensional (nD) Spatio-temporal Systems with Non-causal Spatial Variables
151 p. 2012 [978-83-7842-013-2]

Vol. 20: Michał Doligalski
Behavioral Specification Diversification of Reconfigurable Logic Controllers
108 p. 2012 [978-83-7842-005-7]

Vol. 19: Grzegorz Bazydło
Graphic Specification of Programs for Reconfigurable Logic Controllers Using Unified Modeling Language
121 p. 2012 [978-83-7481-470-6]

Vol. 18: Marek Sawerwain
Selected Topics in Quantum Programming Languages Theory
252 p. 2011 [978-83-7481-457-7]

Vol. 17: Jacek Bieganowski
Synthesis of Microprogram Control Units Oriented Toward Decreasing the Number of Macrocells of Addressing Circuit
105 p. 2011 [978-83-7481-454-6]

Vol. 16: Łukasz Dziekan
Neuro-Fuzzy-Based Takagi-Sugeno Modelling in Fault-Tolerant Control
161 p. 2011 [978-83-7481-427-0]

Vol. 15: Łukasz Hładowski
Efficient Algorithms for Solving Large-scale Computational Control Problems of Repetitive Processes
165 p. 2011 [978-83-7481-415-7]

Vol. 14: Remigiusz Wiśniewski
Synthesis of Compositional Microprogram Control Units for Programmable Devices
153 p. 2009 [978-83-7481-293-1]

Vol. 13: Arkadiusz Bukowiec
Synthesis of Finite State Machines for FPGA Devices Based on Architectural Decomposition
102 p. 2009 [978-83-7481-257-3]

Vol. 12: Małgorzata Kołopieńczyk
Application of Address Converter for Decreasing Memory Size of Compositional Microprogram Control Unit with Code Sharing
96 p. 2008 [83-7481-033-5]

Vol. 11: Bartłomiej Sulikowski
Computational Aspects in Analysis and Synthesis of Repetitive Processes
168 p. 2006 [83-7481-033-5]

Vol. 10: Bartosz Kuczewski
Computational Aspects of Discrimination between Models of Dynamic Systems
158 p. 2006 [83-7481-030-0]

Vol. 9: Marek Kowal
Optimization of Neuro-Fuzzy Structures in Technical Diagnostics Systems
116 p. 2005 [83-89712-88-1]

Vol. 8: Wojciech Paszke
Analysis and Synthesis of Multidimensional System Classes Using Linear Matrix Inequality Methods
188 p. 2005 [83-89712-81-4]

Vol. 7: Piotr Steć
Segmentation of Colour Video Sequences Using the Fast Marching Method
110 p. 2005 [83-89712-47-4]

Vol. 6: Grzegorz Łabiak
The Use of Hierarchical Model of Concurrent Automaton in Digital Controller Design (in Polish)
168 p. 2005 [83-89712-42-3]

Vol. 5: Maciej Patan
Optimal Observation Strategies for Parameter Estimation of Distributed Systems
220 p. 2004 [83-89712-03-2]

Vol. 4: Przemysław Jacewicz
Model Analysis and Synthesis of Complex Physical Systems Using Cellular Automata
134 p. 2003 [83-89321-67-X]

Vol. 3: Agnieszka Węgrzyn
Symbolic Analysis of Binary Control Circuits Using Selected Methods of Petri Net Analysis (in Polish)
125 p. 2003 [83-89321-54-8]

Vol. 2: Grzegorz Andrzejewski
Program Model of Interpreted Petri Net for Digital Microsystem Design (in Polish)
109 p. 2003 [83-89321-53-X]

Vol. 1: Marcin Witczak
Identification and Fault Detection of Non-Linear Dynamic Systems
124 p. 2003 [83-88317-65-2]

Marcin Skobel

Marcin Skobel, Ph.D., was born in 1989. Upon obtaining a M.Sc.. degree in geodesy and cartography in 2013 at the AGH University of Krakow, he started professional practic and in 2018 he completed M.Sc. degree in computer science at the University of Zielona Góra. In 2024, he earned his Ph.D. degree in technical informatics and telecommunications under the supervision of Prof. Marek Kowal. He was a scholarship holder of a research project financed by the National Science Centre under the supervision of Prof. Józef Korbicz. His research interests concern artrificial inteligence methods and image processing and recognition.

ISBN: 978-83-7842-558-8